

Anonymous and Adaptively Secure Revocable IBE with Constant-Size Public Parameters

Jie Chen*, Hoon Wei Lim, San Ling, Le Su and Huaxiong Wang

Abstract

In Identity-Based Encryption (IBE) systems, key revocation is non-trivial. This is because a user's identity is itself a public key. Moreover, the private key corresponding to the identity needs to be obtained from a trusted key authority through an authenticated and secrecy protected channel. So far, there exist only a very small number of revocable IBE (RIBE) schemes that support non-interactive key revocation, in the sense that the user is not required to interact with the key authority or some kind of trusted hardware to renew her private key without changing her public key (or identity). These schemes are either proven to be only selectively secure or have public parameters which grow linearly in a given security parameter. In this paper, we present two constructions of non-interactive RIBE that satisfy all the following three attractive properties: (i) proven to be adaptively secure under the Symmetric External Diffie-Hellman (SXDH) and the Decisional Linear (DLIN) assumptions; (ii) have constant-size public parameters; and (iii) preserve the anonymity of ciphertexts—a property that has not yet been achieved in all the current schemes.

Index Terms

Dual System Encryption, Functional Encryption, Identity-Based Encryption, Key Revocation

I. INTRODUCTION

Identity-based encryption (IBE) allows one's identity to be directly used as a public key [29, 6, 12]. This obviates the need for a public key certificate that attests the binding between the identity and a (seemingly) random key, as in the case of more conventional certificate-based public-key systems. Thus, IBE systems have simpler public key management than that of certificate-based systems. In IBE, however, a private key (corresponding to an identity) needs to be generated by a trusted key authority. This and the fact that a user's identity is itself a public key complicates key renewal or revocation—one cannot simply change her public key, as this changes her identity as well. While there has been a great deal of work on IBE in recent years, see for example [7, 30, 8, 13, 31, 10, 1, 2], not much work has been devoted to *key revocation*.

*Corresponding Author

**All authors are with Division of Mathematical Sciences, School of Physical & Mathematical Sciences, Nanyang Technological University, Singapore. Emails in sequence: s080001@e.ntu.edu.sg, {hoonwei,lingsan}@ntu.edu.sg, lsu1@e.ntu.edu.sg, hxwang@ntu.edu.sg.

One direct way to alleviate the key revocation problem in the IBE setting is to maintain a revocation list by some trusted third party. A sender checks on the trusted third party and just stops to encrypt messages if the corresponding receiver is revoked. However, this direct model requires the trusted third party to keep online in order to respond any sender's real time checking query. To address this problem, one simple solution is to append a validity period to a target identity during encryption [6]. This results in a public key with a limited validity period, and hence, restricting the window of exposure should the corresponding private key is compromised. If the validity period is sufficiently short, one may not require an explicit key revocation mechanism since an exposed private key is of little value to an adversary beyond the specified validity period. However, one major drawback of this approach is that each user has to periodically renew her private key. As a consequence, the key authority's workload increases linearly in the number of non-revoked users. Further, we must ensure that each transmission of a new private key between the key authority and a non-revoked user is performed through some form of authenticated and secure channel. There exist some improved key revocation techniques in the literature, for example [19, 15]. However, they require interactions either between the user and the key authority, as before, or between the user and some kind of trusted hardware. These may not always be practical.

The first non-interactive, revocable IBE (RIBE) scheme that neither presupposes the existence of trusted hardware nor requires a secure channel between the user and the key authority, is due to Boldyreva et al. [5]. Their scheme borrows the concept of fuzzy IBE (FIBE) [28] in which encryption of a message is associated with two "attributes", namely identity of the receiver and time period. The corresponding decryption key is also split into two private components, matching the identity and the time period, respectively. The private component that corresponds to the identity is essentially similar to a regular private key in IBE and it is issued to a user by the key authority through a secure channel. On the other hand, the private key component corresponding to the time period is regarded as a key update and is published by the key authority to all users. (Here the key update is public information and does not require secrecy protection.) Thus, to revoke a user, the key authority simply stops distributing the key update for that user. To reduce the number of key updates to be performed by the key authority, Boldyreva et al. organize and relate users' key updates in a binary tree [3, 22]. Briefly speaking, each node of the tree is assigned some key material and each user is assigned to a leaf node in the tree. Upon registration, the key authority computes and provides the user with a set of distinct private keys (corresponding to its identity) based on the key material for each node in the path from the leaf node corresponding to that user to the root node. To be able to decrypt a ciphertext associated with time t , the user needs just one key update (corresponding to t) computed on the key material associated to any of the nodes on the path from the leaf node of the user to the root node. Thus, when no user is revoked, the key authority publishes just the key update computed on the key material of the root node. When a subset of the users is revoked, the key authority first finds the minimal set of nodes in the tree which contains an ancestor (or, the node itself) among all the leaf nodes corresponding to non-revoked users. The key authority then distributes the key updates for only this set. This way, every update of the revocation list only requires the key authority to perform logarithmic work in the maximal number of users and linear in the number of revoked users.

A. Previous Non-Interactive RIBE Constructions

Although an adaptive-ID secure IBE scheme [30] (which is resilient even against an adversary that is allowed to adaptively select an identity as the attack target based on the responses to the adversary’s queries in a security game) has been in existence for some years, constructing an RIBE scheme with equivalent security guarantee is non-trivial. This is evident from the first RIBE scheme proposed by Boldyreva et al. [5]. Although it is intriguing that their RIBE scheme was constructed from the FIBE scheme of [28] and made clever use of the binary tree technique, the scheme was only proven in the selective-ID model, which is, unfortunately, a rather weak model. This is because the adversary is required to set the challenge identity and time at the beginning of a security game before receiving the relevant public parameters. Nevertheless, Libert and Vergnaud [21] eventually proposed an adaptive-ID secure RIBE scheme using similar key revocation techniques as with [5], and thus solved the problem left open by Boldyreva et al. However, instead of building on FIBE, Libert and Vergnaud adopted a variant [20] of the Waters IBE scheme [30], which is based on partitioning techniques and has a drawback in having public parameters that comprise $\mathcal{O}(\lambda)$ group elements for security parameter λ . Consequently, the Libert and Vergnaud RIBE scheme inherits a similar limitation. Clearly, it is desirable that a scheme has small or constant-size public parameters, secret keys, and ciphertexts, if it were to be deployed in real world applications.

B. RIBE from Dual System Encryption

Moving beyond proving security through the partitioning techniques, Waters proposed the dual system encryption methodology [31], which has been a powerful tool to obtain full security for various classes of functional encryption (FE) [9], such as IBE [31, 17, 16], inner product encryption (IPE) [18], and attribute-based encryption (ABE) [18, 25]. Although there already exist some schemes that achieve full security using the dual system encryption technique, (for example, the HIBE scheme of [17] has been proven to be fully secure by applying this technique to the HIBE scheme of [7]), however, these fully secure schemes typically require relatively large parameters and/or constructed only in the composite order bilinear groups. Thus, in general, the dual system encryption methodology does not always provide generic transformation from selective security to adaptive security without suffering from the mentioned limitations.

In our work, we initially tried to apply the dual system encryption technique to the selective-ID RIBE scheme of [5], however this results an analogous construction and proof to the ABE scheme of [25]. Furthermore, as we illustrate below, such an approach does not enjoy constant-size public parameters and keys.

To see this, we specifically consider the binary-tree key update approach [5, 21] in the setting similar to key-policy ABE.¹ As before, a ciphertext in the RIBE scheme is associated with two attributes: identity id_i and time period t_j . The ciphertext can be decrypted by a user if and only if the user possesses both the private key for identity id_i and the key update for time t_j on some node in the tree. Since the private keys and key updates associated with a specific node are not given to the users simultaneously, collusion among some (non-revoked) users on some

¹The case for ciphertext-policy ABE setting is similar.

attributes (i.e. time attribute) is possible. Hence from the view of ABE, all users can be regarded as “sharing” the same key (or private component) associated with access structure of the form

$$(\text{id}_1 \vee \cdots \vee \text{id}_n) \wedge (t_1 \vee \cdots \vee t_m)$$

on each node in the tree for some integers n and m , but each user is given only some parts of the key for this access structure. That is, the parts of the key that the user gets correspond to access structure $\text{id}_i \wedge (t_1 \vee \cdots \vee t_m)$ if this node is in the path from the leaf node associated with id_i to the root node; while the key updates corresponding to $(t_1 \vee \cdots \vee t_m)$ are given to all users (not necessarily at the same time). Clearly, we require that the private keys are collusion-resistant on different nodes. Moreover, supporting a large universe attribute space is required and can be used to deal with exponential identity spaces in RIBE.

We observe that, however, the adaptively secure ABE schemes of [25] cannot be used directly for our purpose because the resulting RIBE somewhat unexpectedly has private keys and ciphertexts with sizes that grow linearly in the maximal number of users and the size of time space (even though they are polynomial in the security parameter). It turns out that constructing a fully secure RIBE scheme with constant-size public parameters and keys requires additional work.

C. Our Contributions

In this paper, we investigate how to instantiate the Waters dual system encryption methodology with revocable IBE schemes. Particularly, we construct two efficient non-interactive RIBE schemes that are proven to be adaptively secure under the Symmetric External Diffie-Hellman (SXDH) and the Decisional Linear (DLIN) assumptions, respectively.

Our schemes improve the previous work by achieving adaptive security with *constant-size public parameters*. Moreover, our schemes are *anonymous*, namely, preserving the privacy of ciphertext recipients and encryption times. We note that previous RIBE schemes do not consider the anonymity property, an advantage inherited from using the dual pairing vector spaces (DPVS) [23, 24] to achieve orthogonality and entropy-hiding in prime-order groups. Our constructions also make use of the key revocation techniques of [5, 21], namely, we employ binary-tree data structure to achieve key update with logarithmic complexity in the maximal number of users for the key authority.

We give a summary of comparisons between existing and our RIBE schemes in Table I. Here, we use PP to denote public parameters, MK to denote master key, SK to denote private key, KU to denote key update, CT to denote ciphertext, and # pairing to denote the number of pairing computation for decryption. The sizes are in terms of group elements and λ denotes the security parameter.

TABLE I
COMPARISONS BETWEEN EXISTING AND OUR RIBE SCHEMES.

	BGK [5]	LV [21]	Ours	
size of PP	5	$\mathcal{O}(\lambda)$	19	55
size of MK	1	1	19	55
size of SK	2	3	6	9
size of KU	2	3	6	9
size of CT	4	5	6	9
# pairings	4	3	12	18
security	selective	adaptive	adaptive	adaptive
anonymity	No	No	Yes	Yes
assumption	DBDH	mDBDH	SXDH	DLIN

We compare our schemes against Boldyreva et al.’s scheme [5], which is under the Decision Bilinear Diffie-Hellman (DBDH) assumption, and Libert and Vergnaud’s scheme [21], which is under the modified DBDH (mDBDH) assumption. Overall, our schemes are anonymous, adaptively secure, and have constant-size public parameters, at the expense of bigger (but still seems acceptable) sizes in terms of the master key, private key, and key update.

D. Our Approach

In RIBE, different from the standard security game for IBE, the adversary is allowed to query parts of the challenge identities and time periods. Thus, to overcome the problem of increasing sizes of public parameters in the maximal number of users and sizes of the time space as analyzed in the ABE setting, our security proof makes use of two types of nominally semi-functional pairs, while all the previous works based on the dual system encryption methodology, such as [17, 16, 18, 25], require only a single type of nominally semi-functional pair. Moreover, prior to the start of the game, we execute a preliminary game to “locate” the positions of the challenge identities and times. We then transform all the private keys and key updates associated with the non-challenge identities and times, respectively, into nominally semi-functional (we denote by Type I) one by one. We transform the challenge private keys and key updates (or simply keys) into nominally semi-functional (we denote by Type II) node by node at the last step. Note that the distribution of nominally semi-functional pairs of Type I for challenge identities and times can be detected by the adversary that they are different from the distribution of the semi-functional keys and ciphertexts. Moreover, nominally semi-functional pairs of Type II can be only generated for the last remaining keys; in other words, all the other keys must have been already semi-functional. This is why the preliminary game is needed. We also introduce some statistical indistinguishability arguments in our proof to show that the distributions of nominally semi-functional pair of both Types I & II remain the same as the distributions of semi-functional keys and ciphertexts from the adversary’s view. Finally, we arrive at a security game that only requires to generate semi-functional keys and ciphertexts while security can be proved directly.

II. PRELIMINARIES

A. Dual Pairing Vector Spaces

Our constructions are based on dual pairing vector spaces proposed by Okamoto and Takashima [23, 24]. In this paper, we concentrate on the asymmetric version [26]. Particularly, we give a brief description on how to generate random dual orthonormal bases. See [23, 24] for a full definition of dual pairing vector spaces.

Definition 1 (Asymmetric bilinear pairing groups). *Asymmetric bilinear pairing groups* $(q, G_1, G_2, G_T, g_1, g_2, e)$ are a tuple of a prime q , cyclic (multiplicative) groups G_1, G_2 and G_T of order q , $g_1 \neq 1 \in G_1$, $g_2 \neq 1 \in G_2$, and a polynomial-time computable nondegenerate bilinear pairing $e : G_1 \times G_2 \rightarrow G_T$ i.e., $e(g_1^s, g_2^t) = e(g_1, g_2)^{st}$ and $e(g_1, g_2) \neq 1$.

In addition to individual elements of G_1 or G_2 , we will also consider “vectors” of group elements. For $\mathbf{v} = (v_1, \dots, v_n) \in \mathbb{Z}_q^n$ and $g_\beta \in G_\beta$, we write $g_\beta^{\mathbf{v}}$ to denote a n -tuple of elements of G_β for $\beta = 1, 2$:

$$g_\beta^{\mathbf{v}} := (g_\beta^{v_1}, \dots, g_\beta^{v_n}).$$

For any $a \in \mathbb{Z}_q$ and $\mathbf{v}, \mathbf{w} \in \mathbb{Z}_q^n$, we have:

$$g_\beta^{a\mathbf{v}} := (g_\beta^{av_1}, \dots, g_\beta^{av_n}), \quad g_\beta^{\mathbf{v}+\mathbf{w}} := (g_\beta^{v_1+w_1}, \dots, g_\beta^{v_n+w_n}).$$

Then we define

$$e(g_1^{\mathbf{v}}, g_2^{\mathbf{w}}) := \prod_{i=1}^n e(g_1^{v_i}, g_2^{w_i}) = e(g_1, g_2)^{\mathbf{v} \cdot \mathbf{w}}.$$

Here, the dot product is taken modulo q .

Dual Pairing Vector Spaces. For a fixed (constant) dimension n , we choose two random bases $\mathbb{B} := (\mathbf{b}_1, \dots, \mathbf{b}_n)$ and $\mathbb{B}^* := (\mathbf{b}_1^*, \dots, \mathbf{b}_n^*)$ of \mathbb{Z}_q^n , subject to the constraint that they are “dual orthonormal”, meaning that

$$\mathbf{b}_i \cdot \mathbf{b}_j^* = 0 \pmod{q}$$

whenever $i \neq j$, and

$$\mathbf{b}_i \cdot \mathbf{b}_i^* = \psi \pmod{q}$$

for all i , where ψ is a random element of \mathbb{Z}_q . We denote the above algorithm, which generates the dual orthonormal bases, as $\text{Dual}(\cdot)$. Then for generators $g_1 \in G_1$ and $g_2 \in G_2$, we have

$$e(g_1^{\mathbf{b}_i}, g_2^{\mathbf{b}_j^*}) = 1$$

whenever $i \neq j$, where 1 here denotes the identity element in G_T .

B. Complexity Assumptions

To define the SXDH assumption, we first define DDH problems in G_1 and G_2 .

Definition 2 (DDH1: Decisional Diffie-Hellman Assumption in G_1). *Given a group generator \mathcal{G} , we define the following distribution:*

$$\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow_{\mathbf{R}} \mathcal{G},$$

$$a, b, c \leftarrow_{\mathbf{R}} \mathbb{Z}_q,$$

$$D := (\mathbb{G}; g_1, g_2, g_1^a, g_1^b).$$

We assume that for any PPT algorithm \mathcal{A} (with output in $\{0, 1\}$),

$$\text{Adv}_{\mathcal{A}}^{\text{DDH1}}(\lambda) := |\Pr[\mathcal{A}(D, g_1^{ab})] - \Pr[\mathcal{A}(D, g_1^{ab+c})]|$$

is negligible in the security parameter λ .

The dual of the Decisional Diffie-Hellman assumption in G_1 is Decisional Diffie-Hellman assumption in G_2 (denoted as DDH2), which is identical to Definitions 2 with the roles of G_1 and G_2 reversed. We say that:

Definition 3. *The Symmetric External Diffie-Hellman assumption holds if DDH problems are intractable in both G_1 and G_2 .*

The following SXDH-based Subspace assumptions is from [11], which we will use in our security proof.

Definition 4 (DS1: Decisional Subspace Assumption in G_1). *Given a group generator $\mathcal{G}(\cdot)$, define the following distribution:*

$$\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e) \leftarrow_{\mathbf{R}} \mathcal{G}(1^\lambda),$$

$$(\mathbb{B}, \mathbb{B}^*) \leftarrow_{\mathbf{R}} \text{Dual}(\mathbb{Z}_q^N),$$

$$\tau_1, \tau_2, \mu_1, \mu_2 \leftarrow_{\mathbf{R}} \mathbb{Z}_q,$$

$$U_1 := g_2^{\mu_1 \mathbf{b}_1^* + \mu_2 \mathbf{b}_{K+1}^*}, \dots, U_K := g_2^{\mu_1 \mathbf{b}_K^* + \mu_2 \mathbf{b}_{2K}^*},$$

$$V_1 := g_1^{\tau_1 \mathbf{b}_1}, \dots, V_K := g_1^{\tau_1 \mathbf{b}_K},$$

$$W_1 := g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_{K+1}}, \dots, W_K := g_1^{\tau_1 \mathbf{b}_K + \tau_2 \mathbf{b}_{2K}},$$

$$D := (\mathbb{G}; g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_K^*}, g_2^{\mathbf{b}_{2K+1}^*}, \dots, g_2^{\mathbf{b}_N^*},$$

$$g_1^{\mathbf{b}_1}, \dots, g_1^{\mathbf{b}_N}, U_1, \dots, U_K, \mu_2),$$

where K, N are fixed positive integers that satisfy $2K \leq N$. We assume that for any PPT algorithm \mathcal{A} (with output in $\{0, 1\}$),

$$\text{Adv}_{\mathcal{A}}^{\text{DS1}}(\lambda) := |\Pr[\mathcal{A}(D, V_1, \dots, V_K) = 1] - \Pr[\mathcal{A}(D, W_1, \dots, W_K) = 1]|$$

is negligible in the security parameter λ .

Lemma 1. *If the DDH assumption in G_1 holds, then the Subspace assumption in G_1 stated in Definition 4 also holds. More precisely, for any adversary \mathcal{A} against the Subspace assumption in G_1 , there exist probabilistic algorithms \mathcal{B} whose running times are essentially the same as that of \mathcal{A} , such that*

$$\text{Adv}_{\mathcal{A}}^{\text{DS1}}(\lambda) \leq \text{Adv}_{\mathcal{B}}^{\text{DDH1}}(\lambda).$$

The dual of the Subspace assumption in G_1 is Subspace assumption in G_2 (denoted as DS2), which is identical to Definitions 4 with the roles of G_1 and G_2 reversed. Similarly, the Subspace assumption holds in G_2 if the DDH assumption in G_2 holds.

We define the DLIN problem in symmetric bilinear pairing groups (namely $G_1 = G_2$). The DLIN-based Subspace assumptions could be found in [16, 25].

Definition 5 (DLIN: Decisional Linear Assumption). *Given a group generator \mathcal{G} , we define the following distribution:*

$$\begin{aligned} \mathbb{G} &:= (q, G, G_T, g, e) \leftarrow_{\mathcal{R}} \mathcal{G}, \\ a_1, a_2, b_1, b_2, c &\leftarrow_{\mathcal{R}} \mathbb{Z}_q, \\ D &:= (\mathbb{G}; g, g^{a_1}, g^{a_2}, g^{a_1 b_1}, g^{a_2 b_2}). \end{aligned}$$

We assume that for any PPT algorithm \mathcal{A} (with output in $\{0, 1\}$),

$$\text{Adv}_{\mathcal{A}}^{\text{DLIN}}(\lambda) := \left| \Pr[\mathcal{A}(D, g^{b_1+b_2})] - \Pr[\mathcal{A}(D, g_1^{b_1+b_2+c})] \right|$$

is negligible in the security parameter λ .

III. REVOCABLE IBE

We first recall the definition of RIBE and its security from [5] and then define an appropriate security model for our constructions.

Definition 6. *An Identity-Based Encryption with efficient revocation or simply Revocable IBE (RIBE) scheme has seven PPT algorithms Setup, PriKeyGen, KeyUpd, DeckKeyGen, Enc, Dec, and KeyRev with associated message space \mathcal{M} , identity space \mathcal{I} and time space \mathcal{T} . We assume that the size of \mathcal{T} is polynomial in the security parameter. Each algorithm is run by either one of three types of parties—key authority, sender or receiver. Key authority maintains a revocation list RL and state ST. In what follows, an algorithm is called stateful if it updates RL or ST. We treat time as discrete as opposed to continuous.*

- **Setup**($1^\lambda, N_{\max}$) takes as input a security parameter λ and a maximal number of users N_{\max} . It outputs public parameters PP, a master key MK, a revocation list RL (initially empty), and a state ST. (This is run by the key authority.)
- **PriKeyGen**(PP, MK, id, ST) takes as input the public parameters PP, the master key MK, an identity $\text{id} \in \mathcal{I}$, and the state ST. It outputs a private key SK_{id} and an updated state ST. (This is stateful and run by the key authority.)

- **KeyUpd**(PP, MK, t, RL, ST) takes as input the public parameters PP, the master key MK, a key update time $t \in \mathcal{T}$, the revocation list RL, and the state ST. It outputs a key update KU_t . (This is run by the key authority.)
- **DecKeyGen**(SK_{id} , KU_t) takes as input a private key SK_{id} and key update KU_t . It outputs a decryption key $DK_{id,t}$ or a special symbol \perp indicating that id was revoked. (This is deterministic and run by the receiver.)
- **Enc**(PP, id , t, m) takes as input the public parameters PP, an identity $id \in \mathcal{I}$, an encryption time $t \in \mathcal{T}$, and a message $m \in \mathcal{M}$. It outputs a ciphertext $CT_{id,t}$. (This is run by the sender.)
- **Dec**(PP, $DK_{id,t}$, $CT_{id,t}$) takes as input the public parameters PP, a decryption key $DK_{id,t}$, and a ciphertext $CT_{id,t}$. It outputs a message $m \in \mathcal{M}$. (This is deterministic and run by the receiver.)
- **KeyRev**(id , t, RL, ST) takes as input an identity to be revoked $id \in \mathcal{I}$, a revocation time $t \in \mathcal{T}$, the revocation list RL, and the state ST. It outputs an updated revocation list RL. (This is stateful and run by the key authority.)

The consistency condition requires that for all $\lambda \in \mathbb{N}$ and polynomials (in λ) N_{max} , all PP and MK output by setup algorithm **Setup**, all $m \in \mathcal{M}$, $id \in \mathcal{I}$, $t \in \mathcal{T}$ and all possible valid states ST and revocation lists RL, if identity id was not revoked before or, at time t then the following experiment returns 1 except for a negligible probability:

$$\begin{aligned}
 (SK_{id}, ST) &\leftarrow_R \text{PriKeyGen}(PP, MK, id, ST); \\
 KU_t &\leftarrow_R \text{KeyUpd}(PP, MK, t, RL, ST) \\
 DK_{id,t} &\leftarrow \text{DecKeyGen}(SK_{id}, KU_t); \\
 CT_{id,t} &\leftarrow_R \text{Enc}(PP, id, t, m) \\
 \text{If } \text{Dec}(PP, DK_{id,t}, CT_{id,t}) &= m \text{ then return 1 else return 0.}
 \end{aligned}$$

Boldyreva et al. formalized and defined the selective-ID security for RIBE. Their definition captures not only the standard notion of selective-ID security but also takes into account key revocation. The following definition extends the security property expressed in [5] to the adaptive-ID and anonymous setting.

- **Setup**: It is run to generate public parameters PP, a master key MK, a revocation list RL (initially empty), and a state ST. Then PP is given to \mathcal{A} .
- **Query**: \mathcal{A} may adaptively make a polynomial number of queries of the following oracles (the oracles share state):
 - The private key generation oracle **PriKeyGen**(\cdot) takes as input an identity id and runs **PriKeyGen**(PP, MK, id , ST) to return a private key SK_{id} .
 - The key update generation oracle **KeyUpd**(\cdot) takes as input time t and runs **KeyUpd**(PP, MK, t , RL, ST) to return a key update KU_t .
 - The revocation oracle **KeyRev**(\cdot, \cdot) takes as input an identity id and time t and runs **KeyRev**(id , t , RL, ST) to update RL.
- **Challenge**: \mathcal{A} outputs the two challenge pair $(id_{(0)}^*, t_{(0)}^*, m_{(0)}^*), (id_{(1)}^*, t_{(1)}^*, m_{(1)}^*) \in \mathcal{I} \times \mathcal{T} \times \mathcal{M}$. A random bit β is chosen. \mathcal{A} is given **Enc**(PP, $id_{(\beta)}^*, t_{(\beta)}^*, m_{(\beta)}^*$).

- **Guess:** The adversary may continue to make queries of the oracles as in **Query** phase and outputs a bit β' , and succeeds if $\beta' = \beta$.

The following restrictions must always hold:

- 1) **KeyUpd**(\cdot) and **KeyRev**(\cdot, \cdot) can be queried on time which is greater than or equal to the time of all previous queries, i.e., the adversary is allowed to query only in non-decreasing order of time. Also, the oracle **KeyRev**(\cdot, \cdot) cannot be queried at time t if **KeyUpd**(\cdot) was queried on t .
- 2) For $\beta = 0, 1$, if **PriKeyGen**(\cdot) was queried on identity $\text{id}_{(\beta)}$ then **KeyRev**(\cdot, \cdot) must be queried on $(\text{id}_{(\beta)}^*, t)$ for some $t \leq t_{(\beta)}^*$, i.e., identity $\text{id}_{(\beta)}^*$ must be in RL when **KeyUpd**(\cdot) is queried at time $t_{(\beta)}^*$.

For $\beta = 0, 1$ let W_β be the event that the adversary outputs 1 in Experiment β and define

$$\text{Adv}_{\mathcal{A}}^{\text{RIBE}}(\lambda) := |\Pr[W_0] - \Pr[W_1]|.$$

Definition 7. An RIBE scheme is *adaptive-ID secure and anonymous* if for all PPT adversaries \mathcal{A} the function $\text{Adv}_{\mathcal{A}}^{\text{RIBE}}(\lambda)$ is negligible.

Remark: The security notion of *non-anonymous* RIBE is defined as above with restriction that $\text{id}_{(0)}^* = \text{id}_{(1)}^*$ and $t_{(0)}^* = t_{(1)}^*$. On the other hand, if the adversary \mathcal{A} outputs $(\text{id}_{(0)}^*, \text{id}_{(0)}^*)$ and $(\text{id}_{(1)}^*, t_{(1)}^*)$ before the **Setup** phase, it is *selective-ID* security.

IV. CONSTRUCTION FROM SXDH

In this section, we present our first construction of RIBE and its proof of security under the SXDH assumption.

A. The Binary-tree Data Structure

Key revocation in our scheme relies on binary-tree data structure, as with [3, 22, 5, 21]. We denote the binary-tree by BT and its root node by root. If ν is a leaf node then $\text{Path}(\nu)$ denotes the set of nodes on the path from ν to root (both ν and root inclusive). If θ is a non-leaf node then θ_ℓ, θ_r denote the left and right child of θ , respectively. We assume that all nodes in the tree are uniquely encoded as strings, and the tree is defined by all of its node descriptions.

Each user is assigned to a leaf node ν . Upon registration, the key authority provides the user with a set of distinct private keys for each node in $\text{Path}(\nu)$. At time t , the key authority uses an algorithm called **KUNodes** to determine the minimal set Y of nodes in BT such that none of the nodes in RL with corresponding time $\leq t$ (users revoked on or before t) have any ancestor (or, themselves) in the set Y , and all other leaf nodes (corresponding to non-revoked users) have exactly one ancestor (or, themselves) in the set. The **KUNodes** algorithm takes as input a binary tree

BT, a revocation list RL and a time t , and can be formally specified as follows:

```

KUNodes(BT, RL, t)

  X, Y  $\leftarrow \emptyset$ 

   $\forall (\nu_i, t_i) \in \text{RL}$ 

    if  $t_i \leq t$  then add  $\text{Path}(\nu_i)$  to X

   $\forall \theta \in X$ 

    if  $\theta_\ell \notin X$  then add  $\theta_\ell$  to Y

    if  $\theta_r \notin X$  then add  $\theta_r$  to Y

  If  $Y = \emptyset$  then add root to Y

  Return Y

```

The **KUNodes** algorithm marks all the ancestors of revoked nodes as revoked and outputs all the non-revoked children of revoked nodes. The key authority then publishes a key update for all nodes of Y . A user assigned to leaf ν is then able to form an effective decryption key for time t if the set Y contains a node in $\text{Path}(\nu)$. By doing so, every update of the revocation list RL only requires the key authority to perform logarithmic work in the maximal number of users and linear in the number of revoked users.

B. Our Scheme

We now specify our RIBE scheme. We sometimes provide some intuition or remark at the end of an algorithm and this is marked by the symbol “//”.

- **Setup**(λ, N_{max}) On input a security parameter λ , and a maximal number N_{max} of users, and generate a bilinear pairing $\mathbb{G} := (q, G_1, G_2, G_T, g_1, g_2, e)$ for sufficiently large prime order q . Next perform the following steps:
 - 1) Let RL be an empty set and BT be a binary-tree with at least N_{max} leaf nodes, set $ST = BT$.
 - 2) Sample random dual orthonormal bases, $(\mathbb{D}, \mathbb{D}^*) \leftarrow_{\mathcal{R}} \text{Dual}(\mathbb{Z}_q^6)$. Let $\mathbf{d}_1, \dots, \mathbf{d}_6$ denote the elements of \mathbb{D} and $\mathbf{d}_1^*, \dots, \mathbf{d}_6^*$ denote the elements of \mathbb{D}^* . It also picks $\alpha \leftarrow_{\mathcal{R}} \mathbb{Z}_q$ and computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$
 - 3) Output RL, ST, the public parameters

$$\text{PP} := \left\{ \mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3} \right\},$$

and the master key MK

$$\text{MK} := \left\{ \alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*} \right\}.$$

- **PriKeyGen**(PP, MK, id, RL, ST) On input the public parameters PP, the master key MK, an identity id, the revocation list RL, and the state ST, it picks an unassigned leaf node v from BT and stores id in that node. It then performs the following steps:

- 1) For any $\theta \in \text{Path}(v)$, if $\alpha_{\theta,1}, \alpha_{\theta,2}$ are undefined, then pick $\alpha_{\theta,1} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, set $\alpha_{\theta,2} = \alpha - \alpha_{\theta,1}$, and store them in node θ^2 . Pick $r_{\theta,1} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and compute

$$K_{\text{id},\theta} := g_2^{(\alpha_{\theta,1} + r_{\theta,1}\text{id})\mathbf{d}_1^* - r_{\theta,1}\mathbf{d}_2^*}.$$

- 2) Output $\text{SK}_{\text{id}} := \{(\theta, K_{\text{id},\theta})\}_{\theta \in \text{Path}(v)}$, ST.

//The algorithm computes the id-component of the decryption key for all the nodes on the path from the leaf node (corresponding to id) to root.

- **KeyUpd**(PP, MK, t, RL, ST) On input the public parameters PP, the master key MK, a time t, the revocation list RL, and the state ST, it performs the following steps:

- 1) $\forall \theta \in \text{KUNodes}(\text{BT}, \text{RL}, t)$, if $\alpha_{\theta,1}, \alpha_{\theta,2}$ are undefined, then pick $\alpha_{\theta,1} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, set $\alpha_{\theta,2} = \alpha - \alpha_{\theta,1}$, and store them in node θ . Pick $r_{\theta,2} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and compute

$$K_{t,\theta} := g_2^{(\alpha_{\theta,2} + r_{\theta,2}t)\mathbf{d}_1^* - r_{\theta,2}\mathbf{d}_3^*}.$$

- 2) Output $\text{KU}_t := \{(\theta, K_{t,\theta})\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, t)}$.

//The algorithm first finds a minimal set of nodes which contains an ancestor (or, the node itself) of all the non-revoked nodes. Then it computes the t-component of the decryption key for all the nodes in that set.

- **DecKeyGen**($\text{SK}_{\text{id}}, \text{KU}_t$) On input a private secret key $\text{SK}_{\text{id}} := \{(i, K_{\text{id},i})\}_{i \in I}$, $\text{KU}_t := \{(j, K_{t,j})\}_{j \in J}$ for some set of nodes I, J, it runs the following steps:

- 1) $\forall (i, K_{\text{id},i}) \in \text{SK}_{\text{id}}, (j, K_{t,j}) \in \text{KU}_t$, if $\exists (i, j)$ s.t. $i = j$ then $\text{DK}_{\text{id},t} \leftarrow (K_{\text{id},i}, K_{t,j})$; else (if SK_{id} and KU_t do not have any node in common) $\text{DK}_{\text{id},t} \leftarrow \perp$.

- 2) Output $\text{DK}_{\text{id},t}$.

- **Enc**(PP, id, t, m) On input the public parameters PP, an identity id, a time $t \in \mathbb{Z}_q^n$, and a message m, pick $z \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and forms the ciphertext as

$$\text{CT}_{\text{id},t} := \left\{ C := m \cdot (g_T^\alpha)^z, \quad C_0 := g_1^{z(\mathbf{d}_1 + \text{id}\mathbf{d}_2 + t\mathbf{d}_3)} \right\}.$$

- **Dec**(PP, $\text{DK}_{\text{id},t}$, $\text{CT}_{\text{id},t}$) On input the public parameters PP, a decryption key $\text{DK}_{\text{id},t} := (K_{\text{id},\theta}, K_{t,\theta})$, and a ciphertext $\text{CT}_{\text{id},t} := (C, C_0)$, it computes the message as

$$m := C / (e(C_0, K_{\text{id},\theta}) \cdot e(C_0, K_{t,\theta})).$$

- **KeyRev**(id, t, RL, ST) On input an identity id, a time t, the revocation list RL, and the state ST, the algorithm adds (id, t) to RL for all nodes ν associated with identity id and returns RL.

This ends the description of our scheme.

²To avoid having to store $\alpha_{\theta,1}, \alpha_{\theta,2}$ for each node, the authority can derive them from a pseudo-random function of using a shorter seed and re-compute them when necessary.

Correctness: Observe that

$$\begin{aligned}
& e(C_0, K_{id, \theta}) \\
&= e(g_1^{z(\mathbf{d}_1 + id\mathbf{d}_2 + t\mathbf{d}_3)}, g_2^{(\alpha_{\theta,1} + r_{\theta,1}id)\mathbf{d}_1^* - r_{\theta,1}\mathbf{d}_2^*}) \\
&= e(g_1, g_2)^{\alpha_{\theta,1}z\mathbf{d}_1 \cdot \mathbf{d}_1^*} \cdot e(g_1, g_2)^{zr_{\theta,1}id\mathbf{d}_1 \cdot \mathbf{d}_1^* - zr_{\theta,1}id\mathbf{d}_2 \cdot \mathbf{d}_2^*} \\
&= e(g_1, g_2)^{\alpha_{\theta,1}z\mathbf{d}_1 \cdot \mathbf{d}_1^*}.
\end{aligned}$$

Similarly, $e(C_0, K_{t, \theta}) = e(g_1, g_2)^{\alpha_{\theta,2}z\mathbf{d}_1 \cdot \mathbf{d}_1^*}$. The message is recovered as:

$$\begin{aligned}
& C / e(C_0, K_{id, \theta}) \cdot e(C_0, K_{t, \theta}) \\
&= m \cdot (e(g_1, g_2)^{\alpha\mathbf{d}_1 \cdot \mathbf{d}_1^*})^z / e(g_1, g_2)^{\alpha z\mathbf{d}_1 \cdot \mathbf{d}_1^*} \\
&= m.
\end{aligned}$$

C. Proof of Security

Statistical Indistinguishability Lemmas: We require the following two lemmas, which are derived from [26], for our security proofs.

Lemma 2. For $p \in \mathbb{Z}_q$, let

$$C_p := \{(\mathbf{x}, \mathbf{v}) | \mathbf{x} \cdot \mathbf{v} = p, \mathbf{0} \neq \mathbf{x}, \mathbf{0} \neq \mathbf{v} \in \mathbb{Z}_q^n\}.$$

For all $(\mathbf{x}, \mathbf{v}) \in C_p$, for all $(\mathbf{z}, \mathbf{w}) \in C_p$, and $\mathbf{A} \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{n \times n}$ (\mathbf{A} is invertible with overwhelming probability),

$$\Pr[\mathbf{x}\mathbf{A}^\top = \mathbf{z} \wedge \mathbf{v}\mathbf{A}^{-1} = \mathbf{w}] = \frac{1}{\#C_p}.$$

Lemma 3. For $p_1, p_2 \in \mathbb{Z}_q$, let

$$C_{p_1, p_2} := \left\{ (\mathbf{x}, \mathbf{v}_1, \mathbf{v}_2) \mid \mathbf{x} \neq \mathbf{0}, \mathbf{x} \cdot \mathbf{v}_1 = p_1, \mathbf{x} \cdot \mathbf{v}_2 = p_2 \right\}$$

where $\mathbf{x}, \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}_q^n$, $\{\mathbf{v}_1, \mathbf{v}_2\}$ are linearly independent over \mathbb{Z}_q . For all $(\mathbf{x}, \mathbf{v}_1, \mathbf{v}_2) \in C_{p_1, p_2}$, for all $(\mathbf{z}, \mathbf{w}_1, \mathbf{w}_2) \in C_{p_1, p_2}$, and $\mathbf{A} \leftarrow_{\mathbb{R}} \mathbb{Z}_q^{n \times n}$ (\mathbf{A} is invertible with overwhelming probability),

$$\Pr[\mathbf{x}\mathbf{A}^\top = \mathbf{z} \wedge \mathbf{v}_1\mathbf{A}^{-1} = \mathbf{w}_1 \wedge \mathbf{v}_2\mathbf{A}^{-1} = \mathbf{w}_2] = \frac{1}{\#C_{p_1, p_2}}.$$

The following theorem shows that our RIBE scheme is indeed adaptively secure and anonymous.

Theorem 1. *The RIBE scheme is adaptively secure and anonymous under the SXDH assumption. More precisely, for any adversary \mathcal{A} against the RIBE scheme, there exist probabilistic algorithms*

$$\begin{aligned}
& \mathcal{B}_0, \\
& \{\mathcal{B}_{\kappa_1, \kappa_2}\}_{\kappa_1=1, \dots, q_{n_1}, \kappa_2=1, \dots, \lceil \log N_{max} \rceil}, \\
& \{\mathcal{B}_{\kappa_1, \kappa_2}\}_{\kappa_1=q_{n_1}+1, \dots, q_{n_1}+q_{n_2}+1, \kappa_2=1, \dots, N_{max}}, \\
& \{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}\}_{\kappa_2=1, \dots, 4N_{max}}
\end{aligned}$$

whose running times are essentially the same as that of \mathcal{A} , such that

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{RIBE}}(\lambda) &\leq (q_{n_1} q_{n_2})^2 \cdot \left(\text{Adv}_{\mathcal{B}_0}^{\text{DDH1}}(\lambda) + \sum_{\kappa_1=1}^{q_{n_1}} \sum_{\kappa_2=1}^{\lceil \log N_{max} \rceil} \text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DDH2}}(\lambda) + \sum_{\kappa_1=q_{n_1}+1}^{q_{n_2}} \sum_{\kappa_2=1}^{N_{max}} \text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DDH2}}(\lambda) \right. \\ &\quad \left. + \sum_{\kappa_2=1}^{4N_{max}} \text{Adv}_{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}}^{\text{DDH2}}(\lambda) + \frac{6(q_{n_1} \lceil \log N_{max} \rceil + q_{n_2} N_{max}) + 32N_{max} + 6}{q} \right) \end{aligned}$$

where $q_{n_1}, q_{n_2} \geq 4$ are the maximum number of \mathcal{A} 's private key and key update queries respectively.

Proof: We adopt the dual system encryption methodology by Waters [31] to prove the security of our RIBE scheme. We use the concepts of *semi-functional ciphertexts* and *semi-functional keys* in our proof and provide algorithms that generate them. Particularly, we define two types of semi-functional keys: *semi-functional private keys* (for identity) and *semi-functional key updates* (for time). We note that the algorithms (we specify below) are only provided for definitional purposes, and are not part of the RIBE system. In particular, they do not need to be efficiently computable from the public parameters and the master key.

PriKeyGenSF The algorithm picks $r_{\theta,1}, \nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1}$ randomly from \mathbb{Z}_q and forms a semi-functional private key for node θ as

$$\mathbf{K}_{\text{id},\theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,1} + r_{\theta,1} \text{id}) \mathbf{d}_1^* - r_{\theta,1} \mathbf{d}_2^* + [\nu_{\theta,4,1} \mathbf{d}_4^* + \nu_{\theta,5,1} \mathbf{d}_5^* + \nu_{\theta,6,1} \mathbf{d}_6^*]}. \quad (1)$$

KeyUpdSF The algorithm picks $r_{\theta,2}, \nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2}$ randomly from \mathbb{Z}_q and forms a semi-functional updated key for node θ as

$$\mathbf{K}_{\text{t},\theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,2} + r_{\theta,2} \text{t}) \mathbf{d}_1^* - r_{\theta,2} \mathbf{d}_3^* + [\nu_{\theta,4,2} \mathbf{d}_4^* + \nu_{\theta,5,2} \mathbf{d}_5^* + \nu_{\theta,6,2} \mathbf{d}_6^*]}. \quad (2)$$

EncryptSF The algorithm picks $z, \chi_4, \chi_5, \chi_6$ randomly from \mathbb{Z}_q and forms a semi-functional ciphertext as

$$\text{CT}_{\text{id},\text{t}}^{(\text{SF})} := \left\{ \mathbf{C} := \mathbf{m} \cdot (g_T^\alpha)^z, \mathbf{C}_0 := g_1^{z(\mathbf{d}_1 + \text{id} \mathbf{d}_2 + \text{t} \mathbf{d}_3) + (\chi_4 \mathbf{d}_4 + \chi_5 \mathbf{d}_5 + \chi_6 \mathbf{d}_6)} \right\}. \quad (3)$$

We call a private key or key update semi-functional if all its parts are semi-functional, which are denoted as

$$\begin{aligned} \text{SK}_{\text{id}}^{(\text{SF})} &:= \{(\theta, \mathbf{K}_{\text{id},\theta}^{(\text{SF})})\}_{\theta \in \text{Path}(v)} \\ \text{KU}_{\text{t}}^{(\text{SF})} &:= \{(\theta, \mathbf{K}_{\text{t},\theta}^{(\text{SF})})\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t})}. \end{aligned}$$

We observe that a normal ciphertext $\text{CT}_{\text{id},\text{t}}$ can be decrypted by a semi-functional key pair $(\mathbf{K}_{\text{id},\theta}^{(\text{SF})}, \mathbf{K}_{\text{t},\theta}^{(\text{SF})})$ on some node θ , because $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ are orthogonal to all of the vectors in exponent of \mathbf{C}_0 , and hence have no effect on decryption. Similarly, decryption of a semi-functional ciphertext $\text{CT}_{\text{id},\text{t}}^{(\text{SF})}$ by a normal key pair $(\mathbf{K}_{\text{id},\theta}, \mathbf{K}_{\text{t},\theta})$ on some node θ will also succeed because $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$ are orthogonal to all of the vectors in the exponent of the key. When both the ciphertext and key pair on some node are semi-functional, the result of $e(\mathbf{C}_0^{(\text{SF})}, \mathbf{K}_{\text{id},\theta}^{(\text{SF})}) \cdot e(\mathbf{C}_0^{(\text{SF})}, \mathbf{K}_{\text{t},\theta}^{(\text{SF})})$ will have an additional term, namely

$$e(g_1, g_2)^{\sum_{i=4}^6 (\nu_{\theta,i,1} + \nu_{\theta,i,2}) \chi_i \mathbf{d}_i^* \cdot \mathbf{d}_i} = e(g_1, g_2)^{\sum_{i=4}^6 (\nu_{\theta,i,1} + \nu_{\theta,i,2}) \chi_i \psi}.$$

Decryption will then fail unless $\sum_{i=4}^6 (\nu_{\theta,i,1} + \nu_{\theta,i,2}) \chi_i \psi \equiv 0 \pmod{q}$. If this modular equation holds, we say that the private key, key update and ciphertext pair is *nominally semi-functional*. In our security proof, there are two types of nominally semi-functional pairs:

Nominally semi-functional pair of Type I

$$\begin{aligned} K_{\text{id},\theta}^{(\text{SF})} &:= g_2^{(\alpha_{\theta,1} + r_{\theta,1} \text{id}) \mathbf{d}_1^* - r_{\theta,1} \mathbf{d}_2^* + [\nu_{\theta,4,1} \text{id} \mathbf{d}_4^* - \nu_{\theta,4,1} \mathbf{d}_5^*]}, \\ K_{\text{t},\theta}^{(\text{SF})} &:= g_2^{(\alpha_{\theta,2} + r_{\theta,2} \text{t}) \mathbf{d}_1^* - r_{\theta,2} \mathbf{d}_3^* + [\nu_{\theta,4,2} \text{t} \mathbf{d}_4^* - \nu_{\theta,4,2} \mathbf{d}_6^*]}, \\ \text{CT}_{\text{id},\text{t}}^{(\text{SF})} &:= \left\{ C := m \cdot (g_T^\alpha)^z, C_0 := g_1^{z(\mathbf{d}_1 + \text{id} \mathbf{d}_2 + \text{t} \mathbf{d}_3) + [\chi_4 (\mathbf{d}_4 + \text{id} \mathbf{d}_5 + \text{t} \mathbf{d}_6)]} \right\}, \end{aligned}$$

where $r_{\theta,1}, \nu_{\theta,4,1}, r_{\theta,2}, \nu_{\theta,4,2}, z, \chi_4 \leftarrow_{\mathbb{R}} \mathbb{Z}_q$.

Nominally semi-functional pair of Type II

$$\begin{aligned} K_{\text{id},\theta}^{(\text{SF})} &:= g_2^{(\alpha_{\theta,1} + r_{\theta,1} \text{id}) \mathbf{d}_1^* - r_{\theta,1} \mathbf{d}_2^* + [(\alpha_{\theta} + \nu_{\theta,4,1} \text{id}) \mathbf{d}_4^* - \nu_{\theta,4,1} \mathbf{d}_5^*]}, \\ K_{\text{t},\theta}^{(\text{SF})} &:= g_2^{(\alpha_{\theta,2} + r_{\theta,2} \text{t}) \mathbf{d}_1^* - r_{\theta,2} \mathbf{d}_3^* + [(-\alpha_{\theta} + \nu_{\theta,4,2} \text{t}) \mathbf{d}_4^* - \nu_{\theta,4,2} \mathbf{d}_6^*]}, \\ \text{CT}_{\text{id},\text{t}}^{(\text{SF})} &:= \left\{ C := m \cdot (g_T^\alpha)^z, C_0 := g_1^{z(\mathbf{d}_1 + \text{id} \mathbf{d}_2 + \text{t} \mathbf{d}_3) + [\chi_4 (\mathbf{d}_4 + \text{id} \mathbf{d}_5 + \text{t} \mathbf{d}_6)]} \right\}, \end{aligned}$$

where $\alpha_{\theta}, r_{\theta,1}, \nu_{\theta,4,1}, r_{\theta,2}, \nu_{\theta,4,2}, z, \chi_4 \leftarrow_{\mathbb{R}} \mathbb{Z}_q$.

Note that nominally semi-functional pair of Type I is used to transform the non-challenge private key and key update queries into semi-functional ones while Type II is for the challenge private key and key update queries.

Assume that a probabilistic polynomial-time adversary \mathcal{A} makes at most q_{n_1} private key queries $\text{id}_1, \dots, \text{id}_{q_{n_1}}$ and q_{n_2} key update queries $\text{t}_1, \dots, \text{t}_{q_{n_2}}$. Since there are many types of adversaries according to whether the challenges $\text{id}_{(0)}^*, \text{id}_{(1)}^*, \text{t}_{(0)}^*, \text{t}_{(1)}^*$ being queried and the restriction of queries, in order to simplify and unify reduction, we add four dumb queries $\text{id}_{q_{n_1}+1}, \text{id}_{q_{n_1}+2}, \text{t}_{q_{n_2}+1}, \text{t}_{q_{n_2}+2}$ (the keys for these queries will not be given to \mathcal{A}), which makes the challenge identities $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ and times $\text{t}_{(0)}^*, \text{t}_{(1)}^*$ be included in the $q_{n_1} + 2$ private key queries and the $q_{n_2} + 2$ key update queries. For any adversary, we use values φ_1, φ_2 ($0 < \varphi_1 < \varphi_2 < q_{n_1} + 2$) to indicate the positions of $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ being queried, namely either the φ_1 -th or φ_2 -th query is $\text{id}_{(0)}^*$ and the other is $\text{id}_{(1)}^*$. Similarly, we use values φ_3, φ_4 ($0 < \varphi_3 < \varphi_4 < q_{n_2} + 2$) to indicate the positions of $\text{t}_{(0)}^*, \text{t}_{(1)}^*$ being queried.

Our proof of security consists of the following sequence of games between the adversary \mathcal{A} and challengers.

- $\text{Game}_{\text{Real}}$: is the real security game.
- $\text{Game}_{\text{Real}'}$: is a preliminary game, which is the same as $\text{Game}_{\text{Real}}$ except that the challenger picks $\phi_1, \phi_2 \leftarrow_{\mathbb{R}} [q_{n_1} + 2]$ ($0 < \phi_1 < \phi_2 < q_{n_1} + 2$) and $\phi_3, \phi_4 \leftarrow_{\mathbb{R}} [q_{n_2} + 2]$ ($0 < \phi_3 < \phi_4 < q_{n_2} + 2$) before setup, and the game is aborted if $\phi_i \neq \varphi_i$ for any $i \in [4]$.

//Guess the positions of the challenge identities $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ and times $\text{t}_{(0)}^*, \text{t}_{(1)}^*$. If the guess is incorrect then the game aborts. Re-write

$$\begin{aligned} \Gamma_1 &:= \{\text{id}'_1, \dots, \text{id}'_{q_{n_1}}\} = \{\text{id}_1, \dots, \text{id}_{q_{n_1}+2}\} \setminus \{\text{id}_{\varphi_1}, \text{id}_{\varphi_2}\} \\ \Gamma_2 &:= \{\text{t}'_1, \dots, \text{t}'_{q_{n_2}}\} = \{\text{t}_1, \dots, \text{t}_{q_{n_2}+2}\} \setminus \{\text{t}_{\varphi_3}, \text{t}_{\varphi_4}\}. \end{aligned}$$

- Game_0 : is the same as $\text{Game}_{\text{Real}'}$ except that the challenge ciphertext is semi-functional.
- $\text{Game}_{\kappa_1, \kappa_2}$: for κ_1 from 1 to q_{n_1} , for κ_2 from 0 to $\lceil \log N_{\max} \rceil$, $\text{Game}_{\kappa_1, \kappa_2}$ is the same as Game_0 except that the first $\kappa_1 - 1$ private keys and the first κ_2 components of the κ_1 -th private key for Γ_1 are semi-functional and the remaining keys are normal.

//Transform all private keys into semi-functional ones (one by one and node by node) except the ϕ_1 -th and ϕ_2 -th private queries. Namely, the private keys for the challenge identities $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ (if queried) are still normal. Note that the number of nodes associated with a private key is $\lceil \log N_{\max} \rceil$. Moreover $\text{Game}_{1,0}$ and Game_0 , $\text{Game}_{\kappa_1, \lceil \log N_{\max} \rceil}$ and $\text{Game}_{\kappa_1+1,0}$ are identical.

- $\text{Game}_{\kappa_1, \kappa_2}$: for κ_1 from $q_{n_1} + 1$ to q_{n_2} , for κ_2 from 0 to N_{\max} , $\text{Game}_{\kappa_1, \kappa_2}$ is the same as $\text{Game}_{q_{n_1}, \lceil \log N_{\max} \rceil}$ (namely all private keys for Γ_1 are semi-functional) except that the first $\kappa_1 - q_{n_1} - 1$ key updates and the first κ_2 components of the $(\kappa_1 - q_{n_1})$ -th key update for Γ_2 are semi-functional and the remaining key updates are normal.

//Transform all key updates into semi-functional ones (one by one and node by node) except the ϕ_3 -th and ϕ_4 -th key update queries. Namely, the key updates for the challenge times $t_{(0)}^*, t_{(1)}^*$ (if queried) are still normal. Note that a key update for a time updates at most N_{\max} nodes. Moreover, $\text{Game}_{q_{n_1}, \lceil \log N_{\max} \rceil}$ and $\text{Game}_{q_{n_1}+1,0}$, $\text{Game}_{\kappa_1, N_{\max}}$ and $\text{Game}_{\kappa_1+1,0}$ are identical.

- $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2}$: for κ_2 from 0 to $4N_{\max}$, $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ is the same as $\text{Game}_{q_{n_1}+q_{n_2}, N_{\max}}$ (namely all private keys for Γ_1 and key updates for Γ_2 are semi-functional) except that the ϕ_1, ϕ_2 -th private keys, the ϕ_3, ϕ_4 -th key updates for the first κ_2 nodes are semi-functional and the remaining keys are normal.

//Transform the φ_1, φ_2 -th private key and the φ_3, φ_4 -th key update queries into semi-functional ones (node by node). Note that there are at most $2^{\lceil \log N_{\max} \rceil} (\leq 4N_{\max})$ nodes in the binary tree. Moreover, $\text{Game}_{q_{n_1}+q_{n_2}, N_{\max}}$ and $\text{Game}_{q_{n_1}+q_{n_2}+1,0}$ are identical, namely all keys are semi-functional in $\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{\max}}$.

- $\text{Game}_{\text{Final}}$: is the same as $\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{\max}}$, except that the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random identity in \mathbb{Z}_q a random time in \mathbb{Z}_q . We denote the challenge ciphertext in $\text{Game}_{\text{Final}}$ as $\text{CT}_{\text{id}_{(R)}, t_{(R)}}^{(R)}$.

We prove the following lemmas to show the above games are indistinguishable. The advantage gap between $\text{Game}_{\text{Real}}$ and Game_0 is bounded by the advantage of the Subspace assumption in G_1 . Additionally, we require a statistical indistinguishability argument to show that the distribution of the challenge ciphertext remains the same from the adversary's view. Similarly, the advantage gap between any two consecutive games of $\text{Game}_{1,1}$ to $\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{\max}}$ is bounded by the advantage of Subspace assumption in G_2 . Finally, we statistically transform $\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{\max}}$ to $\text{Game}_{\text{Final}}$ in one step, i.e., we show the joint distributions of parameters in these two games are equivalent from the adversary's view.

We let $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}}}$ denote an adversary \mathcal{A} 's advantage in the real game.

Lemma 4. For any adversary \mathcal{A} , $\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}}}(\lambda) \leq (q_{n_1} q_{n_2})^2 \cdot \text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}'}}(\lambda)$.

Proof: Since $\phi_1, \phi_2, \phi_3, \phi_4$ are uniformly and independently generated, which are hidden from the adversary

\mathcal{A} 's view. The game is non-aborted with probability

$$\frac{4}{(q_{n_1} + 2)(q_{n_1} + 1)(q_{n_2} + 2)(q_{n_2} + 1)}.$$

Thus,

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}}}(\lambda) &= \frac{(q_{n_1} + 2)(q_{n_1} + 1)(q_{n_2} + 2)(q_{n_2} + 1)}{4} \cdot \text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}'}}(\lambda) \\ &\leq (q_{n_1} q_{n_2})^2 \cdot \text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}'}}(\lambda). \end{aligned}$$

■

Lemma 5. *Suppose that there exists an adversary \mathcal{A} where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\text{Real}'}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_0}(\lambda)| = \epsilon$. Then there exists an algorithm \mathcal{B}_0 such that $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon + \frac{2}{q}$, with $K = 3$ and $N = 6$.*

Proof: \mathcal{B}_0 is given

$$D := (\mathbb{G}; g_2^{\mathbf{b}_1^*}, g_2^{\mathbf{b}_2^*}, g_2^{\mathbf{b}_3^*}, g_1^{\mathbf{b}_1}, \dots, g_1^{\mathbf{b}_6}, U_1, U_2, U_3, \mu_2).$$

along with T_1, T_2, T_3 . We require that \mathcal{B}_0 decides whether T_1, T_2, T_3 are distributed as $g_1^{\tau_1 \mathbf{b}_1}, g_1^{\tau_1 \mathbf{b}_2}, g_1^{\tau_1 \mathbf{b}_3}$ or $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_4}, g_1^{\tau_1 \mathbf{b}_2 + \tau_2 \mathbf{b}_5}, g_1^{\tau_1 \mathbf{b}_3 + \tau_2 \mathbf{b}_6}$.

\mathcal{B}_0 simulates $\text{Game}_{\text{Real}'}$ or Game_0 with \mathcal{A} , depending on the distribution of T_1, T_2, T_3 . To compute the public parameters and master key, \mathcal{B}_0 chooses a random invertible matrix $\mathbf{A} \in \mathbb{Z}_q^{3 \times 3}$ (\mathbf{A} is invertible with overwhelming probability if it is uniformly picked) and implicitly sets dual orthonormal bases \mathbb{D}, \mathbb{D}^* to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, & \mathbf{d}_2 &:= \mathbf{b}_2, & \mathbf{d}_3 &:= \mathbf{b}_3, & (\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) &:= (\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6)\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, & \mathbf{d}_2^* &:= \mathbf{b}_2^*, & \mathbf{d}_3^* &:= \mathbf{b}_3^*, & (\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*) &:= (\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that \mathbb{D}, \mathbb{D}^* are properly distributed, and reveal no information about \mathbf{A} . Moreover, \mathcal{B}_0 cannot generate $g_2^{\mathbf{d}_4^*}, g_2^{\mathbf{d}_5^*}, g_2^{\mathbf{d}_6^*}$, but these will not be needed for creating normal private keys and key updates. \mathcal{B}_0 chooses random value $\alpha \in \mathbb{Z}_q$ and computes $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. It then gives \mathcal{A} the public parameters

$$\text{PP} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}.$$

The master key

$$\text{MK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}$$

is known to \mathcal{B}_0 , which allows \mathcal{B}_0 to respond to all of \mathcal{A} 's queries by calling the normal private keys, key updates, and key revocation algorithms.

\mathcal{A} sends \mathcal{B}_0 two pairs $(\text{id}_{(0)}^*, t_{(0)}^*, m_{(0)}^*)$ and $(\text{id}_{(1)}^*, t_{(1)}^*, m_{(1)}^*)$. \mathcal{B}_0 chooses a random bit $\beta \in \{0, 1\}$ and encrypts $m_{(\beta)}^*$ under $(\text{id}_{(\beta)}^*, t_{(\beta)}^*)$ as follows:

$$C := m_{(\beta)}^* \cdot \left(e(T_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = m_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad C_0 := T_1(T_2)^{\text{id}_{(\beta)}^*}(T_3)^{t_{(\beta)}^*},$$

where \mathcal{B}_0 has implicitly set $z := \tau_1$. It gives the ciphertext (C, C_0) to \mathcal{A} .

Now, if T_1, T_2, T_3 are equal to $g_1^{\tau_1 \mathbf{b}_1}, g_1^{\tau_1 \mathbf{b}_2}, g_1^{\tau_1 \mathbf{b}_3}$, then this is a properly distributed normal encryption of $m_{(\beta)}^*$. In this case, \mathcal{B}_0 has properly simulated $\text{Game}_{\text{Real}'}$. If T_1, T_2, T_3 are equal to $g_1^{\tau_1 \mathbf{b}_1 + \tau_2 \mathbf{b}_4}, g_1^{\tau_1 \mathbf{b}_2 + \tau_2 \mathbf{b}_5}, g_1^{\tau_1 \mathbf{b}_3 + \tau_2 \mathbf{b}_6}$ instead, then the ciphertext element C_0 has an additional term of

$$\tau_2 \mathbf{b}_4 + \text{id}_{(\beta)}^* \tau_2 \mathbf{b}_5 + \mathbf{t}_{(\beta)}^* \tau_2 \mathbf{b}_6$$

in its exponent. The coefficients here in the basis $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ form the vector $(\tau_2, \text{id}_{(\beta)}^* \tau_2, \mathbf{t}_{(\beta)}^* \tau_2)$. To compute the coefficients in the basis $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$, we multiply the matrix \mathbf{A}^{-1} by the transpose of this vector, obtaining $\tau_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top$. Since \mathbf{A} is random (everything else given to \mathcal{A} has been distributed independently of \mathbf{A}), these coefficients are uniformly random except with probability $2/q$ (namely, the cases τ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) defined in Equation 3 is the zero vector) from Lemma 2. Therefore, in this case, \mathcal{B}_0 has properly simulated Game_0 . This allows \mathcal{B}_0 to leverage \mathcal{A} 's advantage ϵ between $\text{Game}_{\text{Real}'}$ and Game_0 to achieve an advantage $\epsilon + \frac{2}{q}$ against the Subspace assumption in G_1 , namely $\text{Adv}_{\mathcal{B}_0}^{\text{DS1}}(\lambda) = \epsilon + \frac{2}{q}$. ■

Lemma 6. For κ_1 from 1 to q_{n_1} , for κ_2 from 0 to $\lceil \log N_{max} \rceil$, suppose that there exists an adversary \mathcal{A} where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa_1, \kappa_2-1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa_1, \kappa_2}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{\kappa_1, \kappa_2}$ such that $\text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DS2}}(\lambda) = \epsilon + \frac{6}{q}$, with $K = 3$ and $N = 6$.

Proof: $\mathcal{B}_{\kappa_1, \kappa_2}$ is given

$$D := (\mathbb{G}; g_1^{\mathbf{b}_1}, g_1^{\mathbf{b}_2}, g_1^{\mathbf{b}_3}, g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_6^*}, U_1, U_2, U_3, \mu_2)$$

along with T_1, T_2, T_3 . We require that $\mathcal{B}_{\kappa_1, \kappa_2}$ decides whether T_1, T_2, T_3 are distributed as $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$ or $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$.

$\mathcal{B}_{\kappa_1, \kappa_2}$ simulates $\text{Game}_{\kappa_1, \kappa_2}$ or $\text{Game}_{\kappa_1, \kappa_2-1}$ with \mathcal{A} , depending on the distribution of T_1, T_2, T_3 . To compute the public parameters and master key, $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses a random matrix $\mathbf{A} \in \mathbb{Z}_q^{3 \times 3}$ (with all but negligible probability, \mathbf{A} is invertible). We then implicitly set dual orthonormal bases \mathbb{D}, \mathbb{D}^* to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, & \mathbf{d}_2 &:= \mathbf{b}_2, & \mathbf{d}_3 &:= \mathbf{b}_3, & (\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) &:= (\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6) \mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, & \mathbf{d}_2^* &:= \mathbf{b}_2^*, & \mathbf{d}_3^* &:= \mathbf{b}_3^*, & (\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*) &:= (\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*) (\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that \mathbb{D}, \mathbb{D}^* are properly distributed, and reveal no information about \mathbf{A} . $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. \mathcal{B} can give \mathcal{A} the public parameters

$$\text{PP} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}.$$

The master key

$$\text{MK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}$$

is known to $\mathcal{B}_{\kappa_1, \kappa_2}$, which allows $\mathcal{B}_{\kappa_1, \kappa_2}$ to respond to all of \mathcal{A} 's private key and key update queries by calling the normal key generation algorithm. Since $\mathcal{B}_{\kappa_1, \kappa_2}$ also knows $g_2^{\mathbf{d}_4^*}, g_2^{\mathbf{d}_5^*}$, and $g_2^{\mathbf{d}_6^*}$, it can easily produce semi-functional keys. To answer the key queries that \mathcal{A} makes, $\mathcal{B}_{\kappa_1, \kappa_2}$ runs the semi-functional private key and key update generation

algorithm to produce semi-functional keys and gives these to \mathcal{A} . To answer the κ_2 -th component of the κ_1 -th private key for id'_{κ_1} , $\mathcal{B}_{\kappa_1, \kappa_2}$ responds with:

$$K_{\text{id}'_{\kappa_1}, \theta} := (g_2^{\mathbf{b}_1^*})^{\alpha_{\theta,1}} T_1^{\text{id}'_{\kappa_1}} (T_2)^{-1}.$$

This implicitly sets $r_{\theta,1} := \tau_1$. If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then this is a properly distributed normal private key. Otherwise, if T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$, then this is a semi-functional key, whose exponent vector includes

$$\text{id}'_{\kappa_1} \tau_2 \mathbf{b}_4^* - \tau_2 \mathbf{b}_5^* \quad (4)$$

as its component in the span of $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$. To respond to the remaining key queries, $\mathcal{B}_{\kappa_1, \kappa_2}$ simply runs the normal key generation algorithm.

At some point, \mathcal{A} sends $\mathcal{B}_{\kappa_1, \kappa_2}$ two pairs $(\text{id}_{(0)}^*, \mathbf{t}_{(0)}^*, \mathbf{m}_{(0)}^*)$ and $(\text{id}_{(1)}^*, \mathbf{t}_{(1)}^*, \mathbf{m}_{(1)}^*)$. \mathcal{B}_0 chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\mathbf{m}_{(\beta)}^*$ under $(\text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)$ as follows:

$$\mathbf{C} := \mathbf{m}_{(\beta)}^* \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \mathbf{m}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := U_1(U_2)^{\text{id}_{(\beta)}^*} (U_3)^{\mathbf{t}_{(\beta)}^*},$$

where $\mathcal{B}_{\kappa_1, \kappa_2}$ has implicitly set $z := \mu_1$. The “semi-functional part” of the exponent vector here is:

$$\mu_2 \mathbf{b}_4 + \text{id}_{(\beta)}^* \mu_2 \mathbf{b}_5 + \mathbf{t}_{(\beta)}^* \mu_2 \mathbf{b}_6. \quad (5)$$

We observe that if $\text{id}_{(\beta)}^* = \text{id}'_{\kappa_1}$ (which is impossible), then vectors 4 and 5 would be orthogonal, resulting in a nominally semi-functional ciphertext and key pair $(\mathcal{B}_{\kappa_1, \kappa_2})$ can also use T_1, T_2, T_3 to generate private key part for $\mathbf{t}_{(\beta)}^*$ of Type I. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to \mathcal{A} .

We now argue that since $\text{id}_{(\beta)}^* \neq \text{id}'_{\kappa_1}$, in \mathcal{A} 's view the vectors 4 and 5 are distributed as random vectors in the spans of $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$ respectively. To see this, we take the coefficients of vectors 4 and 5 in terms of the bases $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$ and $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$. Using the change of basis matrix \mathbf{A} , we obtain the new coefficients (in vector form) as:

$$\tau_2 \mathbf{A}^\top (\text{id}'_{\kappa_1}, -1, 0)^\top, \mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top.$$

Since the distribution of everything given to \mathbf{A} except for the κ_2 -th component of the κ_1 -th private key $K_{\text{id}'_{\kappa_1}, \theta}$ and the challenge ciphertext $(\mathbf{C}, \mathbf{C}_0)$ is independent of the random matrix \mathbf{A} and $\text{id}_{(\beta)}^* \neq \text{id}'_{\kappa_1}$, we can conclude that these coefficients are uniformly except with probability $4/q$ (namely, the cases μ_2 or τ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) or $(\nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1})$ defined in Equations 3 and 1 is the zero vector) from Lemma 2. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2}$ in this case.

If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then the coefficients of the vector 5 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) defined in Equations 3 is the zero vector) from Lemma 2. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2-1}$ in this case.

In summary, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated either $\text{Game}_{\kappa_1, \kappa_2-1}$ or $\text{Game}_{\kappa_1, \kappa_2}$ for \mathcal{A} , depending on the distribution of T_1, T_2, T_3 . It can therefore leverage \mathcal{A} 's advantage ϵ between these games to obtain an advantage $\epsilon + \frac{6}{q}$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_\kappa}^{\text{DS2}}(\lambda) = \epsilon + \frac{6}{q}$. \blacksquare

Lemma 7. For κ_1 from $q_{n_1} + 1$ to $q_{n_1} + q_{n_2}$, for κ_2 from 0 to N_{max} , suppose that there exists an adversary \mathcal{A} where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa_1, \kappa_2-1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{\kappa_1, \kappa_2}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{\kappa_1, \kappa_2}$ such that $\text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DS2}}(\lambda) = \epsilon + \frac{6}{q}$, with $K = 3$ and $N = 6$.

Proof: $\mathcal{B}_{\kappa_1, \kappa_2}$ is given

$$D := (\mathbb{G}; g_1^{\mathbf{b}_1}, g_1^{\mathbf{b}_2}, g_1^{\mathbf{b}_3}, g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_6^*}, U_1, U_2, U_3, \mu_2)$$

along with T_1, T_2, T_3 . We require that $\mathcal{B}_{\kappa_1, \kappa_2}$ decides whether T_1, T_2, T_3 are distributed as $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$ or $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$.

$\mathcal{B}_{\kappa_1, \kappa_2}$ simulates $\text{Game}_{\kappa_1, \kappa_2}$ or $\text{Game}_{\kappa_1, \kappa_2-1}$ with \mathcal{A} , depending on the distribution of T_1, T_2, T_3 . To compute the public parameters and master key, $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses a random matrix $\mathbf{A} \in \mathbb{Z}_q^{3 \times 3}$ (with all but negligible probability, \mathbf{A} is invertible). We then implicitly set dual orthonormal bases \mathbb{D}, \mathbb{D}^* to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, & \mathbf{d}_2 &:= \mathbf{b}_2, & \mathbf{d}_3 &:= \mathbf{b}_3, & (\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) &:= (\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6)\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, & \mathbf{d}_2^* &:= \mathbf{b}_2^*, & \mathbf{d}_3^* &:= \mathbf{b}_3^*, & (\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*) &:= (\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that \mathbb{D}, \mathbb{D}^* are properly distributed, and reveal no information about \mathbf{A} . $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. \mathcal{B} can give \mathcal{A} the public parameters

$$\text{PP} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}.$$

The master key

$$\text{MK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}$$

is known to $\mathcal{B}_{\kappa_1, \kappa_2}$, which allows $\mathcal{B}_{\kappa_1, \kappa_2}$ to respond to all of \mathcal{A} 's private key and key update queries by calling the normal key generation algorithm. Since $\mathcal{B}_{\kappa_1, \kappa_2}$ also knows $g_2^{\mathbf{d}_4^*}, g_2^{\mathbf{d}_5^*}$, and $g_2^{\mathbf{d}_6^*}$, it can easily produce semi-functional keys. To answer the key queries that \mathcal{A} makes, $\mathcal{B}_{\kappa_1, \kappa_2}$ runs the semi-functional private key and key update generation algorithm to produce semi-functional keys and gives these to \mathcal{A} . To answer the κ_2 -th component of the κ_1 -th private key for id'_{κ_1} , $\mathcal{B}_{\kappa_1, \kappa_2}$ responds with:

$$\text{K}_{\text{id}'_{\kappa_1}, \theta} := (g_2^{\mathbf{b}_1^*})^{\alpha_{\theta, 1}} T_1^{\text{id}'_{\kappa_1}} (T_2)^{-1}.$$

This implicitly sets $r_{\theta, 1} := \tau_1$. If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then this is a properly distributed normal private key. Otherwise, if T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$, then this is a semi-functional key, whose exponent vector includes

$$\text{id}'_{\kappa_1} \tau_2 \mathbf{b}_4^* - \tau_2 \mathbf{b}_5^* \tag{6}$$

as its component in the span of $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$. To respond to the remaining key queries, $\mathcal{B}_{\kappa_1, \kappa_2}$ simply runs the normal key generation algorithm.

At some point, \mathcal{A} sends $\mathcal{B}_{\kappa_1, \kappa_2}$ two pairs $(\text{id}_{(0)}^*, t_{(0)}^*, m_{(0)}^*)$ and $(\text{id}_{(1)}^*, t_{(1)}^*, m_{(1)}^*)$. \mathcal{B}_0 chooses a random bit $\beta \in \{0, 1\}$ and encrypts $m_{(\beta)}^*$ under $(\text{id}_{(\beta)}^*, t_{(\beta)}^*)$ as follows:

$$C := m_{(\beta)}^* \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = m_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad C_0 := U_1(U_2)^{\text{id}_{(\beta)}^*} (U_3)^{t_{(\beta)}^*},$$

where $\mathcal{B}_{\kappa_1, \kappa_2}$ has implicitly set $z := \mu_1$. The “semi-functional part” of the exponent vector here is:

$$\mu_2 \mathbf{b}_4 + \text{id}_{(\beta)}^* \mu_2 \mathbf{b}_5 + t_{(\beta)}^* \mu_2 \mathbf{b}_6. \quad (7)$$

We observe that if $\text{id}_{(\beta)}^* = \text{id}_{\kappa_1}'$ (which is impossible), then vectors 6 and 7 would be orthogonal, resulting in a nominally semi-functional ciphertext and key pair $(\mathcal{B}_{\kappa_1, \kappa_2})$ can also use T_1, T_2, T_3 to generate private key part for $t_{(\beta)}^*$ of Type I. It gives the ciphertext (C, C_0) to \mathcal{A} .

We now argue that since $\text{id}_{(\beta)}^* \neq \text{id}_{\kappa_1}'$, in \mathcal{A} 's view the vectors 6 and 7 are distributed as random vectors in the spans of $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$ respectively. To see this, we take the coefficients of vectors 6 and 7 in terms of the bases $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$ and $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$. Using the change of basis matrix \mathbf{A} , we obtain the new coefficients (in vector form) as:

$$\tau_2 \mathbf{A}^\top (\text{id}_{\kappa_1}', -1, 0)^\top, \mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top.$$

Since the distribution of everything given to \mathbf{A} except for the κ_2 -th component of the κ_1 -th private key $\mathbf{K}_{\text{id}_{\kappa_1}', \theta}$ and the challenge ciphertext (C, C_0) is independent of the random matrix \mathbf{A} and $\text{id}_{(\beta)}^* \neq \text{id}_{\kappa_1}'$, we can conclude that these coefficients are uniformly except with probability $4/q$ (namely, the cases μ_2 or τ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) or $(\nu_{\theta, 4, 1}, \nu_{\theta, 5, 1}, \nu_{\theta, 6, 1})$ defined in Equations 3 and 1 is the zero vector) from Lemma 2. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2}$ in this case.

If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then the coefficients of the vector 7 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) defined in Equations 3 is the zero vector) from Lemma 2. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2-1}$ in this case.

In summary, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated either $\text{Game}_{\kappa_1, \kappa_2-1}$ or $\text{Game}_{\kappa_1, \kappa_2}$ for \mathcal{A} , depending on the distribution of T_1, T_2, T_3 . It can therefore leverage \mathcal{A} 's advantage ϵ between these games to obtain an advantage $\epsilon + \frac{6}{q}$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_\kappa}^{\text{DS2}}(\lambda) = \epsilon + \frac{6}{q}$.

$\mathcal{B}_{\kappa_1, \kappa_2}$ is given

$$D := (\mathbb{G}; g_1^{\mathbf{b}_1}, g_1^{\mathbf{b}_2}, g_1^{\mathbf{b}_3}, g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_6^*}, U_1, U_2, U_3, \mu_2)$$

along with T_1, T_2, T_3 . We require that $\mathcal{B}_{\kappa_1, \kappa_2}$ decides whether T_1, T_2, T_3 are distributed as $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$ or $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$.

$\mathcal{B}_{\kappa_1, \kappa_2}$ simulates $\text{Game}_{\kappa_1, \kappa_2}$ or $\text{Game}_{\kappa_1, \kappa_2-1}$ with \mathcal{A} , depending on the distribution of T_1, T_2, T_3 . To compute the public parameters and master key, $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses a random matrix $\mathbf{A} \in \mathbb{Z}_q^{3 \times 3}$ (with all but negligible probability,

\mathbf{A} is invertible). We then implicitly set dual orthonormal bases \mathbb{D}, \mathbb{D}^* to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, & \mathbf{d}_2 &:= \mathbf{b}_2, & \mathbf{d}_3 &:= \mathbf{b}_3, & (\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) &:= (\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6)\mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, & \mathbf{d}_2^* &:= \mathbf{b}_2^*, & \mathbf{d}_3^* &:= \mathbf{b}_3^*, & (\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*) &:= (\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*)(\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that \mathbb{D}, \mathbb{D}^* are properly distributed, and reveal no information about \mathbf{A} . $\mathcal{B}_{\kappa_1, \kappa_2}$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. \mathcal{B} can give \mathcal{A} the public parameters

$$\text{PP} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}.$$

The master key

$$\text{MK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}$$

is known to $\mathcal{B}_{\kappa_1, \kappa_2}$, which allows $\mathcal{B}_{\kappa_1, \kappa_2}$ to respond to all of \mathcal{A} 's private key and key update queries by calling the normal key generation algorithm. Since $\mathcal{B}_{\kappa_1, \kappa_2}$ also knows $g_2^{\mathbf{d}_4^*}, g_2^{\mathbf{d}_5^*}$, and $g_2^{\mathbf{d}_6^*}$, it can easily produce semi-functional keys. To answer the key queries that \mathcal{A} makes, $\mathcal{B}_{\kappa_1, \kappa_2}$ runs the semi-functional private key and key update generation algorithm to produce semi-functional keys and gives these to \mathcal{A} . To answer the κ_2 -th component of the $(\kappa_1 - q_{n_1})$ -th key update for $\mathbf{t}'_{\kappa_1 - q_{n_1}}$, $\mathcal{B}_{\kappa_1, \kappa_2}$ responds with:

$$\mathbf{K}_{\mathbf{t}'_{\kappa_1 - q_{n_1}}, \theta} := (g_2^{\mathbf{b}_1^*})^{\alpha_{\theta, 2}} T_1^{\mathbf{t}'_{\kappa_1 - q_{n_1}}} (T_3)^{-1}.$$

This implicitly sets $r_{\theta, 2} := \tau_1$. If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then this is a properly distributed normal key update. Otherwise, if T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$, then this is a semi-functional key update, whose exponent vector includes

$$\mathbf{t}'_{\kappa_1 - q_{n_1}} \tau_2 \mathbf{b}_4^* - \tau_2 \mathbf{b}_6^* \quad (8)$$

as its component in the span of $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$. To respond to the remaining key queries, $\mathcal{B}_{\kappa_1, \kappa_2}$ simply runs the normal key generation algorithm.

At some point, \mathcal{A} sends $\mathcal{B}_{\kappa_1, \kappa_2}$ two pairs $(\text{id}_{(0)}^*, \mathbf{t}_{(0)}^*, \mathbf{m}_{(0)}^*)$ and $(\text{id}_{(1)}^*, \mathbf{t}_{(1)}^*, \mathbf{m}_{(1)}^*)$. \mathcal{B}_0 chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\mathbf{m}_{(\beta)}$ under $(\text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)$ as follows:

$$\mathbf{C} := \mathbf{m}_{(\beta)}^* \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \mathbf{m}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := U_1(U_2)^{\text{id}_{(\beta)}^*} (U_3)^{\mathbf{t}_{(\beta)}^*},$$

where $\mathcal{B}_{\kappa_1, \kappa_2}$ has implicitly set $z := \mu_1$. The “semi-functional part” of the exponent vector here is:

$$\mu_2 \mathbf{b}_4 + \text{id}_{(\beta)}^* \mu_2 \mathbf{b}_5 + \mathbf{t}_{(\beta)}^* \mu_2 \mathbf{b}_6. \quad (9)$$

We observe that if $\mathbf{t}_{(\beta)}^* = \mathbf{t}'_{\kappa_1 - q_{n_1}}$ (which is impossible), then vectors 8 and 9 would be orthogonal, resulting in a nominally semi-functional ciphertext and key pair ($\mathcal{B}_{\kappa_1, \kappa_2}$ can also use T_1, T_2, T_3 to generate private key part for $\text{id}_{(\beta)}^*$) of Type I. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to \mathcal{A} .

We now argue that since $\mathbf{t}_{(\beta)}^* \neq \mathbf{t}'_{\kappa_1 - q_{n_1}}$, in \mathcal{A} 's view the vectors 8 and 9 are distributed as random vectors in the spans of $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$ respectively. To see this, we take the coefficients of vectors 8 and 9 in

terms of the bases $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$ and $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$. Using the change of basis matrix \mathbf{A} , we obtain the new coefficients (in vector form) as:

$$\tau_2 \mathbf{A}^\top (\mathbf{t}'_{\kappa_1 - q_{n_1}}, -1, 0)^\top, \mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)^\top.$$

Since the distribution of everything given to \mathbf{A} except for the κ_2 -th component of the $(\kappa_1 - q_{n_1})$ -th key update $\mathbf{K}_{\mathbf{t}'_{\kappa_1 - q_{n_1}}, \theta}$ and the challenge ciphertext (C, C_0) is independent of the random matrix \mathbf{A} and $\mathbf{t}_{(\beta)}^* \neq \mathbf{t}'_{\kappa_1 - q_{n_1}}$, we can conclude that these coefficients are uniformly except with probability $4/q$ (namely, the cases μ_2 or τ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) or $(\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2})$ defined in Equations 3 and 2 is the zero vector) from Lemma 2. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2}$ in this case.

If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then the coefficients of the vector 9 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) defined in Equations 3 is the zero vector) from Lemma 2. Thus, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated $\text{Game}_{\kappa_1, \kappa_2 - 1}$ in this case.

In summary, $\mathcal{B}_{\kappa_1, \kappa_2}$ has properly simulated either $\text{Game}_{\kappa_1, \kappa_2 - 1}$ or $\text{Game}_{\kappa_1, \kappa_2}$ for \mathcal{A} , depending on the distribution of T_1, T_2, T_3 . It can therefore leverage \mathcal{A} 's advantage ϵ between these games to obtain an advantage $\epsilon + \frac{6}{q}$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_\kappa}^{\text{DS2}}(\lambda) = \epsilon + \frac{6}{q}$. ■

Lemma 8. For κ_2 from 0 to $4N_{max}$, suppose that there exists an adversary \mathcal{A} where $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{q_{n_1} + q_{n_2} + 1, \kappa_2 - 1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{q_{n_1} + q_{n_2} + 1, \kappa_2}}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ such that $\text{Adv}_{\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}}^{\text{DS2}}(\lambda) = \epsilon + \frac{8}{q}$, with $K = 3$ and $N = 6$.

Proof: $\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ is given

$$D := (\mathbb{G}; g_1^{\mathbf{b}_1}, g_1^{\mathbf{b}_2}, g_1^{\mathbf{b}_3}, g_2^{\mathbf{b}_1^*}, \dots, g_2^{\mathbf{b}_6^*}, U_1, U_2, U_3, \mu_2)$$

along with T_1, T_2, T_3 . We require that $\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ decides whether T_1, T_2, T_3 are distributed as $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$ or $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$.

$\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ simulates $\text{Game}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ or $\text{Game}_{q_{n_1} + q_{n_2} + 1, \kappa_2 - 1}$ with \mathcal{A} , depending on the distribution of T_1, T_2, T_3 . To compute the public parameters and master key, $\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ chooses a random matrix $\mathbf{A} \in \mathbb{Z}_q^{3 \times 3}$ (with all but negligible probability, \mathbf{A} is invertible). We then implicitly set dual orthonormal bases \mathbb{D}, \mathbb{D}^* to:

$$\begin{aligned} \mathbf{d}_1 &:= \mathbf{b}_1, & \mathbf{d}_2 &:= \mathbf{b}_2, & \mathbf{d}_3 &:= \mathbf{b}_3, & (\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) &:= (\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6) \mathbf{A}, \\ \mathbf{d}_1^* &:= \mathbf{b}_1^*, & \mathbf{d}_2^* &:= \mathbf{b}_2^*, & \mathbf{d}_3^* &:= \mathbf{b}_3^*, & (\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*) &:= (\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*) (\mathbf{A}^{-1})^\top. \end{aligned}$$

We note that \mathbb{D}, \mathbb{D}^* are properly distributed, and reveal no information about \mathbf{A} . $\mathcal{B}_{q_{n_1} + q_{n_2} + 1, \kappa_2}$ chooses random value $\alpha \in \mathbb{Z}_q$ and compute $g_T^\alpha := e(g_1, g_2)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$. \mathcal{B} can give \mathcal{A} the public parameters

$$\text{PP} := \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}.$$

The master key

$$\text{MK} := \{\alpha, g_2^{\mathbf{d}_1^*}, g_2^{\mathbf{d}_2^*}, g_2^{\mathbf{d}_3^*}\}$$

is known to $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$, which allows $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ to respond to all of \mathcal{A} 's private key and key update queries by calling the normal key generation algorithm. Since $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ also knows $g_2^{\mathbf{d}_4^*}$, $g_2^{\mathbf{d}_5^*}$, and $g_2^{\mathbf{d}_6^*}$, it can easily produce semi-functional keys. To answer the key queries that \mathcal{A} makes, $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ runs the semi-functional private key and key update generation algorithm to produce semi-functional keys and gives these to \mathcal{A} .

However, $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ changes the strategy to respond all the components for the κ_2 -th node in the binary tree of private keys and key updates. All key queries for Γ_1 and Γ_2 are similar with the following process except that $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ uses $g_2^{\mathbf{d}_1^*}, \dots, g_2^{\mathbf{d}_6^*}$ to re-randomize the exponents. To answer the component for the challenge identities $\text{id}_{(0)}^*, \text{id}_{(1)}^*$ and times $\mathbf{t}_{(0)}^*, \mathbf{t}_{(1)}^*$ (namely, the ϕ_1, ϕ_2 -th private key and ϕ_3, ϕ_4 -th key update queries) on the κ_2 -th node, $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ picks $\alpha'_{\theta,1}, \alpha''_{\theta,1} \in \mathbb{Z}_q$ and responds with:

$$\begin{aligned} \mathbf{K}_{\text{id}_{(0)}^*, \theta} &:= g_2^{\alpha'_{\theta,1} \mathbf{b}_1^*} (T_1^{\mathbf{b}_1^*})^{\alpha''_{\theta,1}} T_1^{r'_{\theta,1} \text{id}_{(0)}^*} (T_2)^{-r'_{\theta,1}}, \\ \mathbf{K}_{\mathbf{t}_{(0)}^*, \theta} &:= g_2^{(\alpha - \alpha'_{\theta,1})} (T_1^{\mathbf{b}_1^*})^{-\alpha''_{\theta,1}} T_1^{r'_{\theta,2} \mathbf{t}_{(0)}^*} (T_2)^{-r'_{\theta,2}}, \\ \mathbf{K}_{\text{id}_{(1)}^*, \theta} &:= g_2^{\alpha'_{\theta,1} \mathbf{b}_1^*} (T_1^{\mathbf{b}_1^*})^{\alpha''_{\theta,1}} T_1^{r''_{\theta,1} \text{id}_{(1)}^*} (T_2)^{-r''_{\theta,1}}, \\ \mathbf{K}_{\mathbf{t}_{(1)}^*, \theta} &:= g_2^{(\alpha - \alpha'_{\theta,1})} (T_1^{\mathbf{b}_1^*})^{-\alpha''_{\theta,1}} T_1^{r''_{\theta,2} \mathbf{t}_{(1)}^*} (T_2)^{-r''_{\theta,2}}, \end{aligned}$$

where $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ implicitly sets $\alpha_{\theta,1} := \alpha'_{\theta,1} + \alpha''_{\theta,1} \tau_1$ and $\alpha_{\theta,2} := \alpha - \alpha'_{\theta,1} - \alpha''_{\theta,1} \tau_1$ (note that $\alpha_{\theta,1} + \alpha_{\theta,2} = \alpha$). Note that from the restriction of queries for the challenge identities and times, only part of the keys are given to \mathcal{A} .

If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then these are properly distributed normal keys. If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^* + \tau_2 \mathbf{b}_4^*}, g_2^{\tau_1 \mathbf{b}_2^* + \tau_2 \mathbf{b}_5^*}, g_2^{\tau_1 \mathbf{b}_3^* + \tau_2 \mathbf{b}_6^*}$, then these are semi-functional keys, whose exponent vector includes

$$(\alpha''_{\theta,1} \tau_2 + \text{id}_{(0)}^* \tau_2 r'_{\theta,1}) \mathbf{b}_4^* - \tau_2 r'_{\theta,1} \mathbf{b}_5^*, \quad (10)$$

$$(-\alpha''_{\theta,1} \tau_2 + \mathbf{t}_{(0)}^* \tau_2 r'_{\theta,2}) \mathbf{b}_4^* - \tau_2 r'_{\theta,2} \mathbf{b}_6^*, \quad (11)$$

$$(\alpha''_{\theta,1} \tau_2 + \text{id}_{(1)}^* \tau_2 r''_{\theta,1}) \mathbf{b}_4^* - \tau_2 r''_{\theta,1} \mathbf{b}_5^*, \quad (12)$$

$$(-\alpha''_{\theta,1} \tau_2 + \mathbf{t}_{(1)}^* \tau_2 r''_{\theta,2}) \mathbf{b}_4^* - \tau_2 r''_{\theta,2} \mathbf{b}_6^*, \quad (13)$$

as its component in the span of $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$ respectively. To respond to the remaining key queries, $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ simply runs the normal key generation algorithm.

At some point, \mathcal{A} sends $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ two challenge pairs $(\text{id}_{(0)}^*, \mathbf{t}_{(0)}^*, \mathbf{m}_{(0)}^*)$ and $(\text{id}_{(1)}^*, \mathbf{t}_{(1)}^*, \mathbf{m}_{(1)}^*)$. $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ chooses a random bit $\beta \in \{0, 1\}$ and encrypts $\mathbf{m}_{(\beta)}^*$ under $(\text{id}_{(\beta)}^*, \mathbf{t}_{(\beta)}^*)$ as follows:

$$\mathbf{C} := \mathbf{m}_{(\beta)}^* \cdot \left(e(U_1, g_2^{\mathbf{b}_1^*}) \right)^\alpha = \mathbf{m}_{(\beta)}^* \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := U_1 (U_2)^{\text{id}_{(\beta)}^*} (U_3)^{\mathbf{t}_{(\beta)}^*},$$

where $\mathcal{B}_{q_{n_1}+q_{n_2}+1,\kappa_2}$ has implicitly set $z := \mu_1$. The “semi-functional part” of the exponent vector here is:

$$\mu_2 \mathbf{b}_4 + \text{id}_{(\beta)}^* \mu_2 \mathbf{b}_5 + \mathbf{t}_{(\beta)}^* \mu_2 \mathbf{b}_6. \quad (14)$$

We observe that $((\mathbf{C}, \mathbf{C}_0), \mathbf{K}_{\text{id}_{(\beta)}^*, \theta}, \mathbf{K}_{\mathbf{t}_{(\beta)}^*, \theta})$ would result in a nominally semi-functional ciphertext and key pair of Type II. It gives the ciphertext $(\mathbf{C}, \mathbf{C}_0)$ to \mathcal{A} .

Since the adversary \mathcal{A} is only allowed to query one of the following sets for the challenge identities and times:

$$\emptyset, \{\text{id}_{(0)}^*\}, \{\text{id}_{(1)}^*\}, \{t_{(0)}^*\}, \{t_{(1)}^*\}, \{\text{id}_{(0)}^*, \text{id}_{(1)}^*\}, \{t_{(0)}^*, t_{(1)}^*\}, \\ \{\text{id}_{(0)}^*, t_{(1)}^*\}, \{\text{id}_{(1)}^*, t_{(0)}^*\},$$

we now argue that in \mathcal{A} 's view the given vectors 10, 11, 12, 13 and 14 are distributed as random vectors in the spans of $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$ respectively. To see this, we take the coefficients of vectors 10, 11, 12, 13 and 14 in terms of the bases $\mathbf{b}_4^*, \mathbf{b}_5^*, \mathbf{b}_6^*$ and $\mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6$ respectively and translate them into coefficients in terms of the bases $\mathbf{d}_4^*, \mathbf{d}_5^*, \mathbf{d}_6^*$ and $\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6$. Using the change of basis matrix \mathbf{A} and statistical indistinguishability lemmas, we obtain new random coefficients (in vector form), which are summarized in the following Table:

Case	Type of Adversary	New Coefficients
1	\emptyset	$\mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top$
2	$\{\text{id}_{(0)}^*\}$	$\mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top$ $\mathbf{A}^\top (\alpha_{\theta,1}' \tau_2 + \text{id}_{(0)}^* \tau_2 r_{\theta,1}', -\tau_2 r_{\theta,1}', 0)^\top$
3	$\{\text{id}_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top$ $\mathbf{A}^\top (\alpha_{\theta,1}' \tau_2 + \text{id}_{(1)}^* \tau_2 r_{\theta,1}'', -\tau_2 r_{\theta,1}'', 0)^\top$
4	$\{t_{(0)}^*\}$	$\mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top$ $\mathbf{A}^\top (-\alpha_{\theta,1}' \tau_2 + t_{(0)}^* \tau_2 r_{\theta,2}', 0, -\tau_2 r_{\theta,2}')^\top$
5	$\{t_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top$ $\mathbf{A}^\top (-\alpha_{\theta,1}' \tau_2 + t_{(1)}^* \tau_2 r_{\theta,2}'', 0, -\tau_2 r_{\theta,2}'')^\top$
6	$\{\text{id}_{(0)}^*, \text{id}_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top$ $\mathbf{A}^\top (\alpha_{\theta,1}' \tau_2 + \text{id}_{(0)}^* \tau_2 r_{\theta,1}', -\tau_2 r_{\theta,1}', 0)^\top$ $\mathbf{A}^\top (\alpha_{\theta,1}' \tau_2 + \text{id}_{(1)}^* \tau_2 r_{\theta,1}'', -\tau_2 r_{\theta,1}'', 0)^\top$
7	$\{t_{(0)}^*, t_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top$ $\mathbf{A}^\top (-\alpha_{\theta,1}' \tau_2 + t_{(0)}^* \tau_2 r_{\theta,2}', 0, -\tau_2 r_{\theta,2}')^\top$ $\mathbf{A}^\top (-\alpha_{\theta,1}' \tau_2 + t_{(1)}^* \tau_2 r_{\theta,2}'', 0, -\tau_2 r_{\theta,2}'')^\top$
8	$\{\text{id}_{(0)}^*, t_{(1)}^*\}$	$\mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top$ $\mathbf{A}^\top (\alpha_{\theta,1}' \tau_2 + \text{id}_{(0)}^* \tau_2 r_{\theta,1}', -\tau_2 r_{\theta,1}', 0)^\top$ $\mathbf{A}^\top (-\alpha_{\theta,1}' \tau_2 + t_{(1)}^* \tau_2 r_{\theta,2}'', 0, -\tau_2 r_{\theta,2}'')^\top$
9	$\{\text{id}_{(1)}^*, t_{(0)}^*\}$	$\mu_2 \mathbf{A}^{-1} (1, \text{id}_{(\beta)}^*, t_{(\beta)}^*)^\top$ $\mathbf{A}^\top (\alpha_{\theta,1}' \tau_2 + \text{id}_{(1)}^* \tau_2 r_{\theta,1}'', -\tau_2 r_{\theta,1}'', 0)^\top$ $\mathbf{A}^\top (-\alpha_{\theta,1}' \tau_2 + t_{(0)}^* \tau_2 r_{\theta,2}', 0, -\tau_2 r_{\theta,2}')^\top$

Since the distribution of everything given to \mathbf{A} except for the coefficients of the vectors in above Table is independent of the random matrix \mathbf{A} , we can conclude that these coefficients are uniformly except with probability

- $2/q$, namely except for the cases:
 - μ_2 defined in Subspace problem is zero,
 - (χ_4, χ_5, χ_6) defined in Equation 3 the zero vector,

from Lemma 2 for Case 1.

- $4/q$, namely except for the cases:

- μ_2 or τ_2 defined in Subspace problem is zero,
- (χ_4, χ_5, χ_6) or $(\nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1})$ or $(\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2})$ defined in Equations 3, 1 and 2 is the zero vector,

from Lemma 3 for Cases 2-5, since $\alpha''_{\theta,1}$ is randomly picked from \mathbb{Z}_q .

- $6/q$, namely except for the cases:

- μ_2 or τ_2 defined in Subspace problem is zero,
- (χ_4, χ_5, χ_6) or $(\nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1})$ or $(\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2})$ defined in Equations 3, 1 and 2 is the zero vector,
- $(\nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1})$ and $(\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2})$ defined in Equations 1 and 2 are linearly dependent,

from Lemma 2 for Cases 6-9, since $\alpha''_{\theta,1}$ is randomly picked from \mathbb{Z}_q and the coefficients of vectors 10, 11, 12, 13 are linearly independent.

Thus, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ has properly simulated $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ in this case.

If T_1, T_2, T_3 are equal to $g_2^{\tau_1 \mathbf{b}_1^*}, g_2^{\tau_1 \mathbf{b}_2^*}, g_2^{\tau_1 \mathbf{b}_3^*}$, then the coefficients of the vector 14 are uniformly except with probability $2/q$ (namely, the cases μ_2 defined in Subspace problem is zero, (χ_4, χ_5, χ_6) defined in Equations 3 is the zero vector) from Lemma 2. Thus, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ has properly simulated $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2-1}$ in this case.

In summary, $\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ has properly simulated either $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2-1}$ or $\text{Game}_{q_{n_1}+q_{n_2}+1, \kappa_2}$ for \mathcal{A} , depending on the distribution of T_1, T_2, T_3 . It can therefore leverage \mathcal{A} 's advantage ϵ between these games to obtain an advantage $\epsilon + \frac{8}{q}$ against the Subspace assumption in G_2 , namely $\text{Adv}_{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}}^{\text{DS2}}(\lambda) = \epsilon + \frac{8}{q}$. ■

Lemma 9. For any adversary \mathcal{A} , $\text{Adv}_{\mathcal{A}}^{\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{max}}}(\lambda) \leq \text{Adv}_{\mathcal{A}}^{\text{Game}_{Final}}(\lambda) + \frac{1}{q}$.

Proof: To prove this lemma, we show the joint distributions of

$$(\text{PP}, \text{CT}_{\text{id}_{(\beta)}^*, \text{t}_{(\beta)}^*}^{(\text{SF})}, \{\text{SK}_{\text{id}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_1}]}, \{\text{KU}_{\text{t}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_2}]})$$

in Game_{ν} and that of

$$(\text{PP}, \text{CT}_{\text{id}_{(\text{R})}, \text{t}_{(\text{R})}}^{(\text{R})}, \{\text{SK}_{\text{id}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_1}]}, \{\text{KU}_{\text{t}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_2}]})$$

in Game_{Final} are equivalent for the adversary's view, where $\text{CT}_{\text{id}_{(\text{R})}, \text{t}_{(\text{R})}}^{(\text{R})}$ is a semi-functional encryption of a random message in G_T and under a random identity $\text{id}_{(\text{R})}$ in \mathbb{Z}_q and a random time $\text{t}_{(\text{R})}$ in \mathbb{Z}_q .

For this purpose, we pick $\mathbf{A} := (\xi_{i,j}) \leftarrow_{\text{R}} \mathbb{Z}_q^{3 \times 3}$ and define new dual orthonormal bases $\mathbb{F} := (\mathbf{f}_1, \dots, \mathbf{f}_6)$, and

$\mathbb{F}^* := (\mathbf{f}_1^*, \dots, \mathbf{f}_6^*)$ as follows:

$$\begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \\ \mathbf{f}_5 \\ \mathbf{f}_6 \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \xi_{1,1} & \xi_{1,2} & \xi_{1,3} & 1 & 0 & 0 \\ \xi_{2,1} & \xi_{2,2} & \xi_{2,3} & 0 & 1 & 0 \\ \xi_{3,1} & \xi_{3,2} & \xi_{3,3} & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \\ \mathbf{d}_4 \\ \mathbf{d}_5 \\ \mathbf{d}_6 \end{pmatrix},$$

$$\begin{pmatrix} \mathbf{f}_1^* \\ \mathbf{f}_2^* \\ \mathbf{f}_3^* \\ \mathbf{f}_4^* \\ \mathbf{f}_5^* \\ \mathbf{f}_6^* \end{pmatrix} := \begin{pmatrix} 1 & 0 & 0 & -\xi_{1,1} & -\xi_{2,1} & -\xi_{3,1} \\ 0 & 1 & 0 & -\xi_{1,2} & -\xi_{2,2} & -\xi_{3,2} \\ 0 & 0 & 1 & -\xi_{1,3} & -\xi_{2,3} & -\xi_{3,3} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{d}_1^* \\ \mathbf{d}_2^* \\ \mathbf{d}_3^* \\ \mathbf{d}_4^* \\ \mathbf{d}_5^* \\ \mathbf{d}_6^* \end{pmatrix}.$$

It is easy to verify that \mathbb{F} and \mathbb{F}^* are also dual orthonormal, and are distributed the same as \mathbb{D} and \mathbb{D}^* .

Then the public parameters, challenge ciphertext, queried private keys and key updates in $\text{Game}_{q_{n_1}+q_{n_2}+1, 4N_{max}}$ are expressed over bases \mathbb{D} and \mathbb{D}^* as

$$\begin{aligned} \text{PP} &:= \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{d}_1}, g_1^{\mathbf{d}_2}, g_1^{\mathbf{d}_3}\}, \\ \text{CT}_{\text{id}_{(\beta)}}^{(\text{SF})} &:= \left\{ \mathbf{C} := m \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := g_1^{z(\mathbf{d}_1 + \text{id}\mathbf{d}_2 + \text{td}_3) + \chi_4 \mathbf{d}_4 + \chi_5 \mathbf{d}_5 + \chi_6 \mathbf{d}_6} \right\}, \\ \left\{ \text{SK}_{\text{id}_\ell}^{(\text{SF})} := \left\{ \left(\theta, \mathbf{K}_{\text{id}_\ell, \theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,1} + r_{\theta,1} \text{id}_\ell) \mathbf{d}_1^* - r_{\theta,1} \mathbf{d}_2^* + \nu_{\theta,4,1} \mathbf{d}_4^* + \nu_{\theta,5,1} \mathbf{d}_5^* + \nu_{\theta,6,1} \mathbf{d}_6^*} \right) \right\}_{\theta \in \text{Path}(v_\ell)} \right\}_{\ell \in [q_{n_1}]}, \\ \left\{ \text{KU}_{\mathbf{t}_\ell}^{(\text{SF})} := \left\{ \left(\theta, \mathbf{K}_{\mathbf{t}_\ell, \theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,2} + r_{\theta,2} \mathbf{t}_\ell) \mathbf{d}_1^* - r_{\theta,2} \mathbf{d}_3^* + \nu_{\theta,4,2} \mathbf{d}_4^* + \nu_{\theta,5,2} \mathbf{d}_5^* + \nu_{\theta,6,2} \mathbf{d}_6^*} \right) \right\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \mathbf{t}_\ell)} \right\}_{\ell \in [q_{n_2}]}. \end{aligned}$$

Then we can express them over bases \mathbb{F} and \mathbb{F}^* as

$$\begin{aligned} \text{PP} &:= \{\mathbb{G}; g_T^\alpha, g_1^{\mathbf{f}_1}, g_1^{\mathbf{f}_2}, g_1^{\mathbf{f}_3}\}, \\ \text{CT}_{\text{id}_{(\beta)}}^{(\text{SF})} &:= \left\{ \mathbf{C} := m \cdot (g_T^\alpha)^z, \quad \mathbf{C}_0 := g_1^{z' \mathbf{f}_1 + z'_2 \mathbf{f}_2 + z'_3 \mathbf{f}_3 + \chi_4 \mathbf{f}_4 + \chi_5 \mathbf{f}_5 + \chi_6 \mathbf{f}_6} \right\}, \\ \left\{ \text{SK}_{\text{id}_\ell}^{(\text{SF})} := \left\{ \left(\theta, \mathbf{K}_{\text{id}_\ell, \theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,1} + r_{\theta,1} \text{id}_\ell) \mathbf{f}_1^* - r_{\theta,1} \mathbf{f}_2^* + \nu'_{\theta,4,1} \mathbf{f}_4^* + \nu'_{\theta,5,1} \mathbf{f}_5^* + \nu'_{\theta,6,1} \mathbf{f}_6^*} \right) \right\}_{\theta \in \text{Path}(v_\ell)} \right\}_{\ell \in [q_{n_1}]}, \\ \left\{ \text{KU}_{\mathbf{t}_\ell}^{(\text{SF})} := \left\{ \left(\theta, \mathbf{K}_{\mathbf{t}_\ell, \theta}^{(\text{SF})} := g_2^{(\alpha_{\theta,2} + r_{\theta,2} \mathbf{t}_\ell) \mathbf{f}_1^* - r_{\theta,2} \mathbf{f}_3^* + \nu'_{\theta,4,2} \mathbf{f}_4^* + \nu'_{\theta,5,2} \mathbf{f}_5^* + \nu'_{\theta,6,2} \mathbf{f}_6^*} \right) \right\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \mathbf{t}_\ell)} \right\}_{\ell \in [q_{n_2}]}. \end{aligned}$$

where

$$\begin{aligned} z'_1 &:= z - \chi_4 \xi_{1,1} - \chi_5 \xi_{2,1} - \chi_6 \xi_{3,1}, \\ z'_2 &:= z \text{id}_{(\beta)}^* - \chi_4 \xi_{1,2} - \chi_5 \xi_{2,2} - \chi_6 \xi_{3,2}, \\ z'_3 &:= z \text{t}_{(\beta)}^* - \chi_4 \xi_{1,3} - \chi_5 \xi_{2,3} - \chi_6 \xi_{3,3} \end{aligned}$$

$$\left\{ \begin{array}{l} \nu'_{\theta,4,1} := \nu_{\theta,4,1} + \xi_{1,1}(\alpha_{\theta,1} + r_{\theta,1} \text{id}_{\ell}) - r_{\theta,1} \xi_{1,2}, \\ \nu'_{\theta,5,1} := \nu_{\theta,5,1} + \xi_{2,1}(\alpha_{\theta,1} + r_{\theta,1} \text{id}_{\ell}) - r_{\theta,1} \xi_{2,2}, \\ \nu'_{\theta,6,1} := \nu_{\theta,6,1} + \xi_{3,1}(\alpha_{\theta,1} + r_{\theta,1} \text{id}_{\ell}) - r_{\theta,1} \xi_{3,2} \end{array} \right\}$$

for $\theta \in \text{Path}(v_{\ell}), \ell \in [q_{n_1}]$,

$$\left\{ \begin{array}{l} \nu'_{\theta,4,2} := \nu_{\theta,4,2} + \xi_{1,1}(\alpha_{\theta,2} + r_{\theta,2} \text{t}_{\ell}) - r_{\theta,2} \xi_{1,3}, \\ \nu'_{\theta,5,2} := \nu_{\theta,5,2} + \xi_{2,1}(\alpha_{\theta,2} + r_{\theta,2} \text{t}_{\ell}) - r_{\theta,2} \xi_{2,3}, \\ \nu'_{\theta,6,2} := \nu_{\theta,6,2} + \xi_{3,1}(\alpha_{\theta,2} + r_{\theta,2} \text{t}_{\ell}) - r_{\theta,2} \xi_{3,3} \end{array} \right\}$$

for $\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t}_{\ell}), \ell \in [q_{n_2}]$, which are all uniformly distributed if (χ_4, χ_5, χ_6) defined in Equation 3 is a non-zero vector since

$$\begin{aligned} &\{\xi_{ij}\}_{i \in [3], j \in [3]}, \\ &\{\{\nu_{\theta,4,1}, \nu_{\theta,5,1}, \nu_{\theta,6,1}\}_{\theta \in \text{Path}(v_{\ell})}\}_{\ell \in [q_{n_1}]}, \\ &\{\{\nu_{\theta,4,2}, \nu_{\theta,5,2}, \nu_{\theta,6,2}\}_{\theta \in \text{KUNodes}(\text{BT}, \text{RL}, \text{t}_{\ell})}\}_{\ell \in [q_{n_2}]} \end{aligned}$$

are all uniformly picked from \mathbb{Z}_q .

In other words, the coefficients $(z, z \text{id}_{(\beta)}^*, z \text{t}_{(\beta)}^*)$ of $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ in the \mathbf{C}_0 term of the challenge ciphertext is changed to random coefficients $(z'_1, z'_2, z'_3) \in \mathbb{Z}_q \times \mathbb{Z}_q \times \mathbb{Z}_q$ of $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$, thus the challenge ciphertext can be viewed as a semi-functional encryption of a random message in G_T and under a random identity in \mathbb{Z}_q and a random time in \mathbb{Z}_q . Moreover, it is not difficult to check that all other coefficients are well distributed. Thus

$$(\text{PP}, \text{CT}_{\text{id}_{(\beta)}^*, \text{t}_{(\beta)}^*}^{(\text{SF})}, \{\text{SK}_{\text{id}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_1}]}, \{\text{KU}_{\text{t}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_2}]})$$

expressed over bases \mathbb{F} and \mathbb{F}^* is properly distributed as

$$(\text{PP}, \text{CT}_{\text{id}_{(\text{R})}, \text{t}_{(\text{R})}}^{(\text{R})}, \{\text{SK}_{\text{id}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_1}]}, \{\text{KU}_{\text{t}_{\ell}}^{(\text{SF})}\}_{\ell \in [q_{n_2}]})$$

in $\text{Game}_{\text{Final}}$.

In the adversary's view, both $(\mathbb{D}, \mathbb{D}^*)$ and $(\mathbb{F}, \mathbb{F}^*)$ are consistent with the same public key. Therefore, the challenge ciphertext and queried secret keys above can be expressed as keys and ciphertext in two ways, in Game_{ν} over bases $(\mathbb{D}, \mathbb{D}^*)$ and in $\text{Game}_{\text{Final}}$ over bases $(\mathbb{F}, \mathbb{F}^*)$. Thus, $\text{Game}_{q_{n_1}+q_{n_1}+1, 4N_{\max}}$ and $\text{Game}_{\text{Final}}$ are statistically indistinguishable. \blacksquare

Lemma 10. For any adversary \mathcal{A} , $\text{Adv}_{\mathcal{A}}^{\text{Game}_{Final}}(\lambda) = 0$.

Proof: The value of β is independent from the adversary's view in Game_{Final} . Hence, $\text{Adv}_{\mathcal{A}}^{\text{Game}_{Final}}(\lambda) = 0$. ■

In Game_{Final} , the challenge ciphertext is a semi-functional encryption of a random message in G_T and under a random identity in \mathbb{Z}_q and a random time in \mathbb{Z}_q , independent of the two messages, the challenge identities, and times provided by \mathcal{A} . Thus, our RIBE scheme is adaptively secure and anonymous. ■

V. CONSTRUCTION FROM DLIN

We use the same binary tree structure mentioned in previous section in our second construction.

A. Our Scheme

Here we provide our second construction of RIBE under the DLIN assumption. Our RIBE scheme is specified as follows:

- **Setup**(λ, N_{max}) On input a security parameter λ , a maximal number N_{max} of users and generate a symmetric bilinear pairing $\mathbb{G} := (q, G, G_T, g, e)$ for sufficiently large prime order q . Next perform the following steps:
 - 1) Let RL be an empty set and BT be a binary-tree with at least N_{max} leaf nodes, set $ST = BT$.
 - 2) Sample random dual orthonormal bases, $(\mathbb{D}, \mathbb{D}^*) \leftarrow_{\mathbb{R}} \text{Dual}(\mathbb{Z}_q^9)$. Let $\mathbf{d}_1, \dots, \mathbf{d}_9$ denote the elements of \mathbb{D} and $\mathbf{d}_1^*, \dots, \mathbf{d}_9^*$ denote the elements of \mathbb{D}^* . It also picks $\alpha \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and computes $g_T^\alpha := e(g, g)^{\alpha \mathbf{d}_1 \cdot \mathbf{d}_1^*}$.
 - 3) Output RL, ST, the public parameters

$$PP := \{\mathbb{G}; g_T^\alpha, g^{\mathbf{d}_1}, \dots, g^{\mathbf{d}_6}\},$$

and the master key MK

$$MK := \{\alpha, g^{\mathbf{d}_1^*}, \dots, g^{\mathbf{d}_6^*}\}.$$

- **PriKeyGen**(PP, MK, id, RL, ST) On input the public parameters PP, the master key MK, an identity id, the revocation list RL, and the state ST, it picks an unassigned leaf node v from BT and stores id in that node. It then performs the following steps:

- 1) For any $\theta \in \text{Path}(v)$, if $\alpha_{\theta,1}, \alpha_{\theta,2}, \alpha_{\theta,3}$ are undefined, then pick $\alpha_{\theta,1}, \alpha_{\theta,3} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, set $\alpha_{\theta,2} = \alpha - \alpha_{\theta,1}$, and store them in node θ . Pick $r_{\theta,1}, r_{\theta,3} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and compute

$$K_{id,\theta} := g^{(\alpha_{\theta,1} + r_{\theta,1}id)\mathbf{d}_1^* - r_{\theta,1}\mathbf{d}_2^* + (\alpha_{\theta,3} + r_{\theta,3}id)\mathbf{d}_4^* - r_{\theta,3}\mathbf{d}_5^*}.$$

- 2) Output $SK_{id} := \{(\theta, K_{id,\theta})\}_{\theta \in \text{Path}(v)}$, ST.

- **KeyUpd**(PP, MK, t, RL, ST) On input the public parameters PP, the master key MK, a time t, the revocation list RL, and the state ST, it performs the following steps:

- 1) $\forall \theta \in \text{KUNodes}(BT, RL, t)$, if $\alpha_{\theta,1}, \alpha_{\theta,2}, \alpha_{\theta,3}$ are undefined, then pick $\alpha_{\theta,1}, \alpha_{\theta,3} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$, set $\alpha_{\theta,2} = \alpha - \alpha_{\theta,1}$, and store them in node θ . Pick $r_{\theta,2}, r_{\theta,4} \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ and compute

$$K_{t,\theta} := g^{(\alpha_{\theta,2} + r_{\theta,2}t)\mathbf{d}_1^* - r_{\theta,2}\mathbf{d}_3^* + (-\alpha_{\theta,3} + r_{\theta,4}t)\mathbf{d}_4^* - r_{\theta,4}\mathbf{d}_6^*}.$$

- 2) Output $KU_t := \{(t, \theta, K_{t,\theta})\}_{\theta \in KUNodes(BT, RL, t)}$.
- **DecKeyGen**(SK_{id}, KU_t) On input a private secret key $SK_{id} := \{(i, K_{id,i})\}_{i \in I}$, $KU_t := \{(j, K_{t,j})\}_{j \in J}$ for some set of nodes I, J , it runs the following steps:
 - 1) $\forall (i, K_{id,i}) \in SK_{id}, (j, K_{t,j}) \in KU_t$, if $\exists (i, j)$ s.t. $i = j$ then $DK_{id,t} \leftarrow (K_{id,i}, K_{t,j})$; else (if SK_{id} and KU_t do not have any node in common) $DK_{id,t} \leftarrow \perp$.
 - 2) Output $DK_{id,t}$.
 - **Enc**(PP, id, t, m) On input the public parameters PP , an identity id , a time $t \in \mathbb{Z}_q^n$, and a message m , it picks $z_1, z_2 \leftarrow_R \mathbb{Z}_q$ and forms the ciphertext as

$$CT_{id,t} := \left\{ C := m \cdot (g_T^\alpha)^{z_1}, C_0 := g^{z_1(d_1 + id d_2 + t d_3) + z_2(d_4 + id d_5 + t d_6)} \right\}.$$
 - **Dec**($PP, DK_{id,t}, CT_{id,t}$) On input the public parameters PP , a decryption key $DK_{id,t} := (K_{id,\theta}, K_{t,\theta})$, and a ciphertext $CT_{id,t} := (C, C_0)$, it computes the message as

$$m := C / (e(C_0, K_{id,\theta}) \cdot e(C_0, K_{t,\theta})).$$
 - **KeyRev**(id, t, RL, ST) On input an identity id , a time t , the revocation list RL , and the state ST , the algorithm adds (id, t) to RL for all nodes ν associated with identity id and returns RL .

Correctness. Observe that

$$\begin{aligned}
 & e(C_0, K_{id,\theta}) \\
 &= e(g^{z_1(d_1 + id d_2 + t d_3) + z_2(d_4 + id d_5 + t d_6)}, g^{(\alpha_{\theta,1} + r_{\theta,1} id) d_1^* - r_{\theta,1} d_2^* + (\alpha_{\theta,3} + r_{\theta,3} id) d_4^* - r_{\theta,3} d_5^*}) \\
 &= e(g, g)^{\alpha_{\theta,1} z_1 d_1 \cdot d_1^* + \alpha_{\theta,3} z_2 d_4 \cdot d_4^*}.
 \end{aligned}$$

Similarly, $e(C_0, K_{t,\theta}) = e(g, g)^{\alpha_{\theta,2} z_1 d_1 \cdot d_1^* - \alpha_{\theta,3} z_2 d_4 \cdot d_4^*}$. Then

$$\begin{aligned}
 & e(C_1, K_{id,\theta}) \cdot e(C_1, K_{t,\theta}) \\
 &= e(g, g)^{\alpha_{\theta,1} z_1 d_1 \cdot d_1^* + \alpha_{\theta,3} z_2 d_4 \cdot d_4^*} \cdot e(g, g)^{\alpha_{\theta,2} z_1 d_1 \cdot d_1^* - \alpha_{\theta,3} z_2 d_4 \cdot d_4^*} \\
 &= g_T^{(\alpha_{\theta,1} + \alpha_{\theta,2}) z_1} \\
 &= (g_T^\alpha)^{z_1}.
 \end{aligned}$$

B. Proof of Security

We show the RIBE scheme is secure by the following theorem, the proof techniques are essentially the same as those for Theorem 1 except that we use the DLIN-based Subspace assumption of [16].

Theorem 2. *The RIBE scheme is adaptively secure and anonymous under the DLIN assumption. More precisely,*

for any adversary \mathcal{A} against the RIBE scheme, there exist probabilistic algorithms

$$\begin{aligned} & \mathcal{B}_0, \\ & \{\mathcal{B}_{\kappa_1, \kappa_2}\}_{\kappa_1=1, \dots, q_{n_1}, \kappa_2=1, \dots, \lceil \log N_{max} \rceil}, \\ & \{\mathcal{B}_{\kappa_1, \kappa_2}\}_{\kappa_1=q_{n_1}+1, \dots, q_{n_1}+q_{n_2}+1, \kappa_2=1, \dots, N_{max}}, \\ & \{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}\}_{\kappa_2=1, \dots, 4N_{max}} \end{aligned}$$

whose running times are essentially the same as that of \mathcal{A} , such that

$$\begin{aligned} \text{Adv}_{\mathcal{A}}^{\text{RIBE}}(\lambda) & \leq (q_{n_1} q_{n_2})^2 \cdot \left(\text{Adv}_{\mathcal{B}_0}^{\text{DLIN}}(\lambda) + \sum_{\kappa_1=1}^{q_{n_1}} \sum_{\kappa_2=1}^{\lceil \log N_{max} \rceil} \text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DLIN}}(\lambda) + \sum_{\kappa_1=q_{n_1}+1}^{q_{n_2}} \sum_{\kappa_2=1}^{N_{max}} \text{Adv}_{\mathcal{B}_{\kappa_1, \kappa_2}}^{\text{DLIN}}(\lambda) \right. \\ & \quad \left. + \sum_{\kappa_2=1}^{4N_{max}} \text{Adv}_{\mathcal{B}_{q_{n_1}+q_{n_2}+1, \kappa_2}}^{\text{DLIN}}(\lambda) + \frac{6(q_{n_1} \lceil \log N_{max} \rceil + q_{n_2} N_{max}) + 32N_{max} + 6}{q} \right) \end{aligned}$$

where $q_{n_1}, q_{n_2} \geq 4$ are the maximum number of \mathcal{A} 's private key and key update queries respectively.

VI. CONCLUSIONS

In this paper, we presented two efficient RIBE schemes under the SXDH and the DLIN assumptions, respectively, which overcome the existing problem of increasing sizes of public parameters. In comparison with the existing schemes of [5, 21], our RIBE schemes are adaptively secure, anonymous and have constant-size public parameters, although they have larger sizes of keys and ciphertexts. Our RIBE schemes can be extended very naturally to obtain revocable IPE schemes with weakly attribute-hiding [24, 26]. Also our techniques can be applied to a more generally setting, for example, the ABE schemes of [26] to obtain adaptively secure revocable ABE schemes.

REFERENCES

- [1] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (h)ibe in the standard model. In *EUROCRYPT*, pages 553–572, 2010.
- [2] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe. In *CRYPTO*, pages 98–115, 2010.
- [3] W. Aiello, S. Lodha, and R. Ostrovsky. Fast digital identity revocation (extended abstract). In *CRYPTO*, pages 137–152, 1998.
- [4] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.
- [5] A. Boldyreva, V. Goyal, and V. Kumar. Identity-based encryption with efficient revocation. In *ACM Conference on Computer and Communications Security*, pages 417–426, 2008.
- [6] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.
- [7] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.

- [8] D. Boneh, C. Gentry, and M. Hamburg. Space-efficient identity based encryption without pairings. In *FOCS*, pages 647–657, 2007.
- [9] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *TCC*, pages 253–273, 2011.
- [10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552, 2010.
- [11] J. Chen, H. W. Lim, S. Ling, H. Wang, and H. Wee. Shorter ibe and signatures via asymmetric pairings. *Pairing*, 2012. To appear, also Cryptology ePrint Archive, Report 2012/224.
- [12] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.
- [13] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.
- [14] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.
- [15] Y. Hanaoka, G. Hanaoka, J. Shikata, and H. Imai. Identity-based hierarchical strongly key-insulated encryption and its application. In *ASIACRYPT*, pages 495–514, 2005.
- [16] A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pages 318–335, 2012.
- [17] A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In *TCC*, pages 455–479, 2010.
- [18] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pages 62–91, 2010.
- [19] B. Libert and J.-J. Quisquater. Efficient revocation and threshold pairing based cryptosystems. In *PODC*, pages 163–171, 2003.
- [20] B. Libert and D. Vergnaud. Towards black-box accountable authority ibe with short ciphertexts and private keys. In *Public Key Cryptography*, pages 235–255, 2009.
- [21] B. Libert and D. Vergnaud. Adaptive-id secure revocable identity-based encryption. In *CT-RSA*, pages 1–15, 2009.
- [22] M. Naor and K. Nissim. Certificate revocation and certificate update. *IEEE Journal on Selected Areas in Communications*, 18(4):561–570, 2000.
- [23] T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Pairing*, pages 57–74, 2008.
- [24] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, pages 214–231, 2009.
- [25] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.

- [26] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. *IACR Cryptology ePrint Archive*, 2010:563, 2010.
- [27] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, pages 195–203, 2007.
- [28] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.
- [29] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
- [30] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.
- [31] B. Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO*, pages 619–636, 2009.